Computation of multivalued traveltimes in three-dimensional heterogeneous media

Dissertation

zur Erlangung des Doktorgrades

der Naturwissenschaften im Fachbereich

Geowissenschaften

der Universität Hamburg

vorgelegt von

Radu Aurel Coman

aus

Făgăraş, Rumänien

Hamburg

2003

Als Dissertation angenommen vom Fachbereich Geowissenschaften der Universität Hamburg

auf Grund der Gutachten

von Prof. Dr. Dirk Gajewski

und Dr. Ekkehart Tessmer

Hamburg, den 09. Juli 2003

Prof. Dr. H. Schleicher

Abstract

In this thesis, I propose several methods for an efficient computation of 2D and 3D traveltimes of transmitted waves in smooth media. Traveltimes are needed in several seismic processing methods such as Kirchhoff prestack and poststack migration, migration velocity analysis, Kirchhoff modelling, or traveltime tomography. The greatest number of traveltimes are needed by 3D prestack Kirchhoff depth migration (PKDM). Despite the continuous increase in computing capacities, the computation of traveltimes for 3D PKDM is still a challenge.

Since the wavefront is usually folded, the same wave may arrive at a subsurface point more than one time, i.e., we have multivalued traveltimes. For efficient computation of multivalued traveltimes in 3D heterogeneous media, I implement a modified 3D wavefront construction (WFC) method, a hybrid method which combines a 3D finite-difference eikonal solver (FDES) with the 3D WFC method, and I develop the wavefront-oriented ray-tracing (WRT) technique (2D and 3D version).

The WFC method starts at the source with few rays which are propagated stepwise through a smooth velocity model. The ray field is examined at each wavefront and a new ray is interpolated between two adjacent rays if certain criteria are satisfied. The traveltimes computed along the rays are used for the estimation of traveltimes on a rectangular grid. This estimation is carried out within a 3D cell, i.e., a region bounded by two consecutive wavefronts and three rays.

This WFC method can be used as a stand-alone method for the computation of multivalued traveltimes or can be combined with a FDES to a hybrid method. The FDES-WFC hybrid method computes the first arrivals by the FDES and the later arrivals by the WFC method.

The WRT technique is related to the WFC methods, but in contrast to the latter the WRT technique inserts a new ray by tracing it directly from the source. This is an approach of higher accuracy than the insertion of a new ray by interpolation on (or near) the wavefront. There are also other differences between WFC methods and the WRT technique. The 3D WRT technique applies a new set of insertion criteria which allows a better control of the ray density. To estimate the traveltimes at gridpoints in the WRT technique, I propose a new approach, which I call the distance-weighted averaging of extrapolated traveltimes.

Contents

List of Figures

Chapter 1

Introduction

The main goal of this work was the development of a method for an efficient computation of multivalued traveltimes in three-dimensional (3D) media with smooth velocity distribution. Under efficient computation I understand a fast computation of accurate traveltimes. The computational speed is important for seismic imaging of complex 3D structures by *prestack Kirchhoff depth migration* (PKDM). Accurate traveltimes are important not only for a good quality of the migrated image, but also for computing migration weights directly from traveltimes (Vanelle and Gajewski, 2002).

PKDM is the most popular technique for imaging of complex 3D geological structures. To date, 3D marine surveys acquired with towed streamers have been prestack migrated almost exclusively using Kirchhoff techniques (Gray et al., 2001).

For Kirchhoff migration, the subsurface is represented on a *migration grid*. A velocity is assigned to each grid point, and each grid point corresponds to a point scatterer. The discretisation interval depends on the resolution of the reflection data under consideration, and usually leads to a spacing of 5-25 m for the migration grid. The amplitudes of the recorded seismic traces are then stacked along the diffraction time curves, which are computed for each scatterer. If the traveltimes are computed for a correct velocity model, then the migrated image corresponds to the geological structure. To save storage space and computational effort, the traveltimes are computed on a *coarse grid*. The traveltimes on the migration grid are obtained during the migration by interpolation (Vanelle and Gajewski, 2002).

For a 3D PKDM the efficiency of traveltime computation is essential (Yilmaz, 2001). The computation of traveltimes by finite-difference (FD) eikonal solvers (FDESs) is fast but most of these methods fail to compute multivalued traveltimes. In case of multivalued traveltimes, the wavefront is folded, i.e., there is more than one ray between the source point and a gridpoint within the region with folded wavefronts. Using only the minimum-traveltime arrivals for PKDM, the energy of unconsidered rays will be lost. As a result the imaged reflector will be displayed with smaller amplitude. It was shown in several

papers that consideration of the maximum energy arrival (e.g., Geoltrain and Brac, 1993; Ettrich and Gajewski, 1996) or of multivalued arrivals (e.g., Operto et al., 2000) increases the quality of the migrated images.

Multivalued traveltimes are usually computed by wavefront construction (WFC) methods (e.g., Vinje et al., 1996a; Ettrich and Gajewski, 1996; Lucio et al., 1996). In these methods, the whole ray field is propagated rather than individual rays. The ray density is evaluated at wavefronts, and if certain insertion criteria are satisfied then a new ray is interpolated in the model. The traveltimes and other ray quantities are estimated within regions bounded by adjacent rays and wavefronts. The main differences between the WFC methods are in (1) the type of insertion criteria, (2) the procedure for the interpolation of new rays, and (3) the approach for the estimation of traveltimes within cells. The WFC methods are considered to be the most suitable methods for the computation of multivalued traveltimes, but their efficiency (accuracy and computational speed) is not sufficient for the increasing amount of seismic data and the required quality of migrated images. A more efficient computation of multivalued traveltime was the aim of my work.

In this thesis, I propose several methods for an efficient computation of multivalued traveltimes. All methods are based on the ray theory, which is presented in Chapter 2. This chapter also includes a review of different methods of traveltime computation. The methods which I propose belong to the larger group of ray-tracing methods. Some general considerations for the numerical implementation of ray-tracing methods are reviewed in Chapter 3.

In Chapter 4, I present my implementation of the wavefront construction (WFC) method. This implementation is based to the WFC methods proposed by Vinje et al. (1996a), Lucio et al. (1996), and Ettrich and Gajewski (1996). Additionally, I use a new approach for the interpolation of new rays in the model. Aspects of this WFC method have been published in Coman and Gajewski (2000b).

For 2D media Ettrich and Gajewski (1997) combine a FDES with a WFC method to a hybrid method. The basic idea of this hybrid method is the computation of first arrivals by a FDES, the automatic detection and bounding of regions where later arrivals occur, and the final application of a WFC method to compute later arrivals. The application of a WFC method is restricted to the bounded regions. In Chapter 5, I present the extention of this hybrid method to 3D media. This 3D hybrid method was published in Coman and Gajewski (2000a).

The strength of the WFC methods is their high computational efficiency and the ability to compute multivalued arrivals. The weak point of the WFC methods is the poor accuracy of the newly interpolated ray. Other aspects which may be improved are the estimation of traveltimes within ray cells, and the control of the ray density by insertion criteria. To remove these weak points, I propose the wavefront-oriented ray-tracing (WRT) technique, which is presented in Chapter 6 for 2D velocity models, and in Chapter 7 for 3D velocity models. In this technique, I insert the new rays by tracing them directly from

the source. This insertion approach increases the accuracy of computed traveltimes and allows a lower overall ray density. The lower ray density requires a new set of insertion criteria and a new approach for the estimation of traveltime within cells. The 2D and 3D WRT techniques have been published in Coman and Gajewski (2002b) and in Coman and Gajewski (2002a).

The main conclusions are summarised in Chapter 8. The notations which are used are given in Appendix A, a list of symbols in Appendix B. Appendix C reviews some theoretical aspects about geometrical spreadings and caustics, while some aspects connected to the numerical implementation of the WRT technique are presented in Appendix D.

Chapter 2

Ray theory

The traveltime computation methods which will be presented in the next chapters are based on the ray theory. The aim of this chapter is to introduce the terminology and the equations which are important for this thesis. For a detailed treatment of the ray theory see for example, Červený (2001); Hubral and Krey (1980); Aki and Richards (1980); Kravtsov and Orlov (1990).

In Section 2.1, the eikonal equation is derived from the elastodynamic equation using a high-frequency (HF) approximation. Solving the eikonal equation by the method of characteristics leads to the kinematic ray tracing (KRT) system (Section 2.2). The eikonal equation and the KRT system are the basis for traveltime computations.

There are different methods which allow the computation of traveltimes (e.g., network ray-tracing methods, finite-difference eikonal solvers, ray-tracing methods). The advantages and disadvantages of these methods are discussed in Section 2.3.

The ray theory presented in this chapter is the basis for my implementation of the WFC method, for the hybrid method, and for the WRT technique. Because these methods are implemented in isotropic media, most of the equations are mentioned only for this type of media. However, these methods can be extended to anisotropic media. To underline the main differences between isotropic and anisotropic ray tracing, the eikonal equation and the KRT system are presented for both types of media.

2.1 Eikonal equation

Consider an anisotropic perfect elastic inhomogeneous medium described by the *elastic tensor* c_{ijkl} (lowercase indices take values 1, 2, 3) and the *density* ρ. The elastic tensor can contain at most 21 independent constants because of symmetry and energy considerations. The number of independent constants depends on the symmetry of the system. An

isotropic medium is described by the *Lamé elastic parameters* λ and μ. Elastic parameters, density and their derivatives are assumed to be continuous functions of coordinates x_i.

The *displacement* $\vec{u}(\vec{x},t)$ in the Cartesian system is described by the *elastodynamic equation* (Červený, 2001, p.13)

$$(c_{ijkl}u_{k,l})_{,j} + f_i = \rho u_{i,tt}\,, \tag{2.1}$$

where t denotes the *time* and f_i the *Cartesian components of the body forces* (also referred to as the source term). Partial derivatives with respect to Cartesian coordinates or time are denoted by a comma (see Appendix A).

Analytical solutions for the elastodynamic equation exist only for very simple models. Currently, numerical solutions are suitable for investigating seismic wave fields in small models (several wavelengths). For prestack depth migration in large 3D models, the numerical solutions of the elastodynamic equation are too costly. For such models it is more efficient to use a HF approximation of the elastodynamic equation.

The HF approximation should be made in media in which the characteristic dimensions of all inhomogeneities (e.g., the radii of curvature of the interface or the thickness of the layers) are considerably larger than the prevailing wavelength of the propagating wave. It is quite common to use the HF approximation even in situations where the above conditions are not fulfilled. This could be useful to understand wave propagation in complex models, but it is necessary to remember that the results may be inaccurate.

The zero-order HF ansatz solution of the elastodynamic equation in case of a harmonic wave reads (Červený, 2001, p.57)

$$\vec{u}(\vec{x},t) = \vec{U}(\vec{x})\,F(t-\tau(\vec{x}))\,, \tag{2.2}$$

where F denotes a HF source signal, $\vec{U}$ the *vectorial amplitude*, and τ the *eikonal*. Synonyms for eikonal are phase function and traveltime. The traveltime is the time which a body wave needs for its propagation between two points, e.g., the source and the receiver. The surface where τ=constant is called wavefront.

Inserting the HF solution (2.2) into the elastodynamic equation (2.1) without body forces ($f_i = 0$) yields

$$N_i(\vec{U})F_{,tt} - M_i(\vec{U})F_{,t} + L_i(\vec{U})F = 0\,, \tag{2.3}$$

where the operators N_i, M_i and L_i are defined in terms of their components in the following way (Červený, 2001, p.62):

$$\begin{aligned} N_i(\vec{U}) &= \Gamma_{ik}U_k - U_i\,, \\ M_i(\vec{U}) &= a_{ijkl}p_jU_{k,l} + \rho^{-1}(\rho a_{ijkl}p_lU_k)_{,j}\,, \\ L_i(\vec{U}) &= (\rho a_{ijkl}U_{k,l})_{,j}\,. \end{aligned} \tag{2.4}$$

Here a_{ijkl} is the *density normalised elastic tensor* ($= c_{ijkl}/\rho$) and

$$\Gamma_{ik} = a_{ijkl} p_j p_l \tag{2.5}$$

will be referred to as *Christoffel matrix* (Červený, 2001, p.22). The vector $\hat{\mathbf{p}}$ is the *slowness* defined as

$$p_j = \tau_{,j}\,. \tag{2.6}$$

Since F is a HF signal

$$|F_{,tt}| \gg |F_{,t}| \gg |F|\,,$$

we can neglect the term that contains F with respect to the terms containing $F_{,t}$ and $F_{,tt}$ in equation (2.3).

To obtain $\tau(\vec{x})$ and $U(\vec{x})$ independent of frequency, equation (2.3) is solved as follows:

$$N_i(\vec{U}) = 0\,, \tag{2.7}$$
$$M_i(\vec{U}) = 0\,. \tag{2.8}$$

The system of equations (2.7) and (2.8) is called the basic system of equations of the ray method. Equation (2.7) will be used to determine the traveltime function $\tau(\vec{x})$ and the polarisation of the amplitude vector $\vec{U}$. The amplitude is determined from equation (2.8) (see Appendix C).

Equations (2.7) can be rewritten as an eigenvalue problem

$$(\Gamma_{ik} - G_m \delta_{ik}) U_k = 0\,, \tag{2.9}$$

where $m = 1,2,3$ and δ_{ik} denotes the *Kronecker symbol*,

$$\delta_{ik} = 1 \text{ for } i = k, \quad \delta_{ik} = 0 \text{ for } i \neq k\,. \tag{2.10}$$

Equation (2.9) is known as the *Christoffel equation*. The matrix Γ_{ik} is positive definite and from this follows that the *eigenvalues* $G_m(x_i, p_i)$ are real and positive. The eigenvalue problem (2.9) represents a system of three equations for U_1, U_2 and U_3, which requires that

$$\det(\Gamma_{ik} - G_m \delta_{ik}) = 0\,. \tag{2.11}$$

The *eigenvectors* $\hat{\mathbf{g}}^{(1)}$, $\hat{\mathbf{g}}^{(2)}$, and $\hat{\mathbf{g}}^{(3)}$ corresponding to the eigenvalues, are mutually orthogonal and can be determined from equation

$$(\Gamma_{ik} - G_m \delta_{ik}) g_k^{(m)} = 0 \tag{2.12}$$

(no summation over m).

Multiplying equation (2.12) by $g_i^{(m)}$ and inserting the expression for Γ_{ik} (equation 2.5) we get

$$G_m = a_{ijkl} p_j^{(m)} p_l^{(m)} g_k^{(m)} g_i^{(m)} \tag{2.13}$$

the *anisotropic eikonal equations*. The eigenvalues for a selected wave satisfy the equation

$$G_m(x_i, p_i) = 1\,. \tag{2.14}$$

In the anisotropic case, there are three different eigenvectors which solve equation (2.11). They correspond to two quasi-shear waves (qS) and a quasi-compressional wave (qP). In the isotropic case, the eigenvectors $\hat{\mathbf{g}}^{(1)}$ and $\hat{\mathbf{g}}^{(2)}$ corresponding to the eigenvalues G_1 and G_2 lie in a plane which is perpendicular on the eigenvector $\hat{\mathbf{g}}^{(3)}$, but the direction of $\hat{\mathbf{g}}^{(1)}$ and $\hat{\mathbf{g}}^{(2)}$ cannot be determined.

In the isotropic case, the expressions for G_m read

$$G_1 = G_2 = v_s^2 \tau_{,i} \tau_{,i}\,, \quad G_3 = v_p^2 \tau_{,i} \tau_{,i}\,, \tag{2.15}$$

where v_s denotes the velocity of S waves and v_p the velocity of P waves. Using $G_m = 1$ and the definition of slowness (2.6), the equations (2.15) read

$$p_i p_i = \frac{1}{v^2}\,, \tag{2.16}$$

where v is either v_p or v_s. This is the *eikonal equation* for isotropic media. The slowness $\hat{\mathbf{p}}$ can be expressed as

$$\hat{\mathbf{p}} = \frac{1}{v}\hat{\mathbf{n}}\,, \tag{2.17}$$

where $\hat{\mathbf{n}}$ is the normal of the wavefront.

2.2 Kinematic ray tracing

The eikonal equation (2.16) in isotropic media can be rewritten as $\mathcal{H}(x_i, p_i) = 0$, with $\mathcal{H}$ as the Hamilton function. This equation represents a partial differential equation of the Hamilton-Jacobi type which are usually solved by applying the method of characteristics. The characteristic system reads (e.g., Kravtsov and Orlov, 1990)

$$\begin{aligned} \frac{\mathrm{d}x_i}{\mathrm{d}v} &= \frac{\partial \mathcal{H}}{\partial p_i}\,, \\ \frac{\mathrm{d}p_i}{\mathrm{d}v} &= -\frac{\partial \mathcal{H}}{\partial x_i}\,, \end{aligned} \tag{2.18}$$

where ν is the parameter variable, which satisfies the equation

$$\frac{d\tau}{d\nu} = p_i \frac{\partial \mathcal{H}}{\partial p_i} . \tag{2.19}$$

Suited ray parameters are, for example, the *traveltime* τ or the *arclength* s along the ray. Selecting the traveltime as the ray parameter and $\mathcal{H} = \frac{1}{2}\left(p_i p_i v^2 - 1\right) = 0$, we transform equation equation (2.19) into the eikonal equation (2.16). The characteristics can be written as

$$\begin{aligned} \frac{dx_i}{d\tau} &= v^2 p_i , \\ \frac{dp_i}{d\tau} &= -\frac{v_{,i}}{v} . \end{aligned} \tag{2.20}$$

The system of equations (2.20) builds the *kinematic ray-tracing (KRT) system*. Solving the six ordinary differential equations of the first order yields the ray trajectory ($x_i = x_i(\tau)$) and the components of the slowness vector at each point of the ray ($p_i = p_i(\tau)$).

The initial conditions for solving the ray-tracing system are given, for example, by the source coordinates and by the slowness components at the source point. The slowness components must satisfy the eikonal equation. Instead of slowness it is possible to use the *take-off angles* γ_1 and γ_2. The quantities γ_i, with $\gamma_3 = \nu$, are usually called the *parameters of the ray*. The ray-tracing system with the initial condition specified above can be solved by standard numerical techniques such as the Runge-Kutta method.

The method of characteristics also leads to a *ray-tracing system for anisotropic media*. This system reads

$$\begin{aligned} \frac{dx_i}{d\tau} &= a_{ijkl} g_j g_k p_l , \\ \frac{dp_i}{d\tau} &= -\frac{1}{2} a_{mjkl,i} g_j g_k p_m p_l . \end{aligned} \tag{2.21}$$

The direction of the vector $dx_i/d\tau$ defines the ray direction and its magnitude is the *ray velocity* or *group velocity*. In anisotropic media the ray direction does not correspond to the slowness direction and the *phase velocity* differs from the ray velocity. Ray tracing in anisotropic media is more costly because of the increased number of operations required to solve (2.21) compared to (2.20). For more details on ray tracing in anisotropic media see e.g., Babich (1961), Červený (1972), Červený et al. (1977), Petrashen and Kashtan (1984), Gajewski and Pšenčik (1987), Farra and Le Bégat (1995).

The problem of ray tracing when the ray is specified by the initial direction and the initial position is called *Cauchy ray tracing* or *initial-value ray tracing*. For practical application,

we often need *two-point ray tracing*, where the ray is specified by the initial and end points of the ray. To find the ray which connects the two points is difficult especially in 3D media. The classical solution is to exploit standard Cauchy ray tracing, taking one of both points as the initial point and iteratively changing the initial direction. This method is usually combined with the paraxial ray approximation.

The eikonal equation and the KRT system are the basis for the computation of traveltimes.

2.3 Traveltime computation methods

The main groups of methods which can be used to compute traveltimes for prestack Kirchhoff depth migration (PKDM) are

- Ray-tracing methods
- Finite-difference eikonal solvers (FDESs)
- Network ray-tracing (NRT) methods

All methods represent the model by a discrete grid of points at which velocities are specified. For 2D media, a comparison of these methods in terms of computational time, accuracy, and memory requirement has been made by Leidenfrost et al. (1999).

Network ray-tracing methods

In the NRT methods each gridpoint is connected by linear segments with a predefined number of other gridpoints. Traveltimes are assigned to these segments. Traveltimes between two arbitrary gridpoints are obtained by summing the traveltime of connections that form the path from the starting point to the endpoint.

The number of possible paths between two points may be very large. According to Fermat's principle, the path with the smallest traveltime is the one which best approximates one of the ray trajectories. The smallest traveltime corresponds to the traveltime of the first arrival. The NRT method was introduced by Nakanishi and Yamaguchi (1986), and the method has been further developed and refined by, for example, Moser (1991), Saito (1989), Klimeš and Kvasnička (1994).

The NRT methods represent a robust way to calculate traveltimes and ray paths of first arrivals. These methods also permit an accurate computation, but they are slow in comparison with other methods (Leidenfrost et al., 1999). Because of the low computational speed, the NRT methods are not a choice for the computation of traveltimes for 3D PKDM. Another drawback of the NRT methods is the difficulty to compute later arrivals.

Finite-difference eikonal solvers

This group of methods solves the eikonal equation by using a finite-difference scheme. The first FDES was proposed by Reshef and Kosloff (1986), who consider only one-way propagation and an expanding half-space scheme. This method was extended to 3D by Reshef (1991). The breakthrough of FDESs was realised by Vidale (1988), who computes 2D first-arrival traveltimes in Cartesian coordinates using an expanding square scheme. The corresponding 3D version apply an expanding cube scheme (Vidale, 1990). The methods proposed by Vidale are very fast, but they are not accurate for large velocity contrasts.

There are several possibilities to overcome this drawback, e.g., by using an expanding wavefront approach (Qin et al., 1992; Sethian and Popovici, 1999), by considering the Huygens' principle to distinguish between different propagation modes, such as transmitted, diffracted or head wave (Podvin and Lecomte, 1991), or by incorporating a correction-by-iteration method (Kim and Cook, 1999).

Other important implementations are FDESs that vectorise (e.g., van Trier and Symes, 1991; Schneider, 1995). The FDESs have been also extended to anisotropic media (e.g., Dellinger, 1991; Eaton, 1993; Lecomte, 1993; Ettrich and Gajewski, 1998; Soukina et al., 2001).

The main drawback of the above mentioned FDESs is that only first-arrival traveltimes are computed. The first arrival can be a head wave, i.e., a wave which is not desired for Kirchhoff migration. An alternative to first-arrival traveltimes is first direct-arrival traveltimes (Mo and Harris, 2002). A direct arrival is a transmitted wave without any part of its propagation being a reflection, a diffraction, or a head wave. The importance of later arrivals was demonstrated by several authors (e.g., Geoltrain and Brac, 1993; Ettrich and Gajewski, 1996). Operto et al. (2000) show the importance of using multivalued arrivals for migration. During the last years, several concepts have been developed to apply FDES for the computation of later arrivals in PKDM (e.g., Bevc, 1995; Benamou, 1996; Ettrich and Gajewski, 1997; Symes, 1998; Sava and Fomel, 2001).

FDESs usually do not allow a flexible choice of the output grid with respect to the grid distance and target region. A FDES grid is finer than the coarse grid which may be used for PKDM, i.e., a lot of computational effort is wasted. In contrast to FDESs, the ray-tracing methods allow a flexible choice of the output grid.

Ray-tracing methods

In the **standard two-point ray-tracing method**, the initial-value ray-tracing technique is applied within an iterative loop to find the ray passing through a predefined gridpoint. To accelerate the convergence of the iterations, the paraxial ray approximation may be used.

Because two-point ray tracing requires rays passing through or near specified gridpoints, this method is not sufficiently efficient for 3D PKDM.

The **controlled initial-value ray tracing** (Bulant, 1999) is a shooting method which does not have the above mentioned requirement. This method decomposes the ray-parameter domain γ_I into homogeneous domains bounded by rays of equal history (e.g., rays which have passed the same interfaces and have the same KMAH index). This decomposition is made on the final surface, which may be represented by a horizontal plane in the target region. The homogeneous subdomains are then triangulated, i.e., ray tubes bounded by three adjacent rays are introduced. These ray tubes can then be decomposed into ray cells, separated by consecutive wavefronts. Inside each cell, the traveltimes may be calculated by interpolation or by using paraxial ray approximations.

Multivalued traveltime tables are usually computed by ray methods which propagate a ray field rather than a single ray. To increase the efficiency of these methods, adjacent rays are grouped into ray tubes, the ray density of the ray field is evaluated at certain positions, and if necessary new rays are inserted. The ray field may be examined at constant depth levels (Lambaré et al., 1992), at interfaces (Åstebøl, 1994), at the final surface (Bulant, 1999), or at the wavefronts. The last approach was introduced by Vinje et al. (1993) and appears to be the most powerful one. Since then, several improvements and other related implementations of the method have been proposed (Sun, 1992; Vinje et al., 1996a,b; Lambaré et al., 1996; Ettrich and Gajewski, 1996; Lucio et al., 1996; Moser and Pajchel, 1997; Vinje, 1997; Bulant and Klimeš, 1999; Buske and Kästner, 1999). Methods which examine the ray field at wavefronts are called **wavefront construction methods**.

Both the wavefront construction (WFC) methods and the controlled initial-value ray-tracing method decompose the computational model into ray cells. However, the WFC methods evaluate the ray density at each wavefront, while the controlled initial-value ray-tracing method evaluates the ray density only at a final surface. Consequently, the ray field in the WFC methods is more adequate for the subsequent traveltime interpolation. The WFC methods are considered to be the most suitable methods for PKDM.

The methods which I propose belong to the group of ray-tracing methods. Some general considerations for the numerical implementation of ray-tracing methods are reviewed in the next chapter.

Chapter 3

General considerations

3.1 Overview

In this chapter, I review some general considerations for the practical implementation of the ray-tracing methods which are described in the next chapters. This review is based on published results.

For the representation of the model (Section 3.2), I analyse two alternatives: the blocky model and the discrete grid model. For migration, the discrete grid representation seems to be the better alternative. A discrete grid representation needs a smooth velocity model.

The smoothing of the velocity model and the influence of smoothing on the efficiency of ray tracing and on the quality of migration results is discussed in Section 3.3. A higher degree of smoothing makes ray tracing faster, but the quality of the migrated image is degraded. I smooth the velocity model by a three point average operator.

There are several possibilities for the estimation of model parameters at arbitrary points. In Section 3.4, I compare the linear interpolation with the interpolation by cardinal splines. The linear interpolation is faster but requires more memory. If 3D dynamic ray-tracing (DRT) is used, then the interpolation by cardinal splines seems to be a better alternative.

In Section 3.5, I outline the important aspects for the integration of the KRT system in WFC methods. The efficiency and accuracy of the integration depends on the integration method and on the integration steps. For integration, I use a fourth-order Runge-Kutta method.

3.2 Representation of the model

In this section, I discuss the representation of the input model for ray tracing. First, I review which petro-physical parameters are needed, then I compare the blocky model

representation with the discrete grid representation.

Ray tracing in isotropic media needs three input parameters: two elastic constants (λ, μ) and the density, or two velocities (v_P, v_S) and the density. If our aim is the migration of compressional reflection data, then the P-wave velocity is sufficient as input parameter.

Ray tracing in general anisotropic media needs 21 elastic constants and the density. In these media we have three waves (one qP and two qS). Usually, each of these waves propagates with its own velocity. For each wave we have a phase velocity (or wave velocity) and a group velocity (or ray velocity). The phase velocity is perpendicular to the surface of constant phase (wavefront) and points in the direction of the slowness vector. The wavefront expands in the direction of the phase velocity. The group velocity has the direction of the ray. The energy is transported in the direction of the group velocity.

Velocity terminology is often misused and causes much confusion. An overview which clarifies the meaning of different definitions of velocities is given, for example, by Sheriff and Geldart (1995). In anisotropic media, the phase velocity and the group velocity do not coincide and both of them depend on direction (in isotropic media, the phase velocity and the group velocity coincide and do not dependent on direction). In anisotropic media, instead of velocities the elements of the elastic tensor have to be given as input parameters.

Two different subsurface representations can be used: the blocky model representation and the discrete grid model representation. To ensure the applicability of ray tracing, the blocky model needs smooth interfaces while the discrete grid model needs a smooth velocity distribution.

The blocky model requires the application of Snell's law and allows the computation of reflection and transmission coefficients. PKDM does not need reflected traveltimes and reflection coefficients, but the transmission coefficients could be used in amplitude-preserving migration for the correction of amplitudes by the transmission losses. On the other hand, the transmission losses can be also estimated to some degree in a gridbased representation of the model (Wang, 2000).

The blocky model seems to be the more natural representation, but there are some severe drawbacks:

- It is difficult to construct complex 3D blocky models.
- Smoothing the interfaces may have more negative influence than smoothing the velocity distribution (Lailly and Sinoquet, 1996).
- The ray-tracing implementation in blocky models is more complex.

Mispel and Hanitzsch (1996) compared blocky models with gridbased models in relation to the quality of migrated images. They have shown that a smooth gridbased velocity

model is sufficient to provide an accurate migration result, and that this is true even if the velocity model is not very accurate (5 % error). The discrete grid representation was also applied with success for the migration in complex models (e.g., Ettrich and Gajewski, 1996; Operto et al., 2000).

Lailly and Sinoquet (1996) compared blocky models with gridbased models in relation to tomographic methods and migration velocity analysis. They have shown that an inaccurate blocky model can make interpretation impossible, while when using the smooth gridbased model the interpretation is possible.

Taking into consideration the above mentioned aspects, I decided to implement the WFC method for a discrete grid representation of the model.

The influence of smoothing the velocity model on the efficiency of ray tracing is discussed in the next section.

3.3 Smoothing the velocity model

In this section, I refer to smoothing as a spatial low-pass filtering process of the velocity model which changes the values of input quantities at gridpoints (e.g., convolutional-type operators, approximating splines). The velocity model can also be smoothed by using interpolating splines, but this type of smoothing does not change the input quantities at gridpoints. Interpolating splines may be used to estimate the input quantities at arbitrary points. This estimation will be discussed in the next section.

Smoothing is important not only for the ray-tracing procedure, but also for the quality of migrated images. Smoothing (1) guarantees the applicability of ray tracing, (2) increases computational speed, and (3) reduces the memory requirement for input quantities. On the other hand, too much smoothing degrades the quality of the migrated image.

Smoothing is a process which naturally occurs during the propagation of seismic waves in the earth. The effective velocity for real wave propagation might be some average within the first Fresnel volume. As the Fresnel volume is connected with the direction of the propagation and the frequency of the wavelet, the effective local medium would be anisotropic and frequency dependent (Gajewski et al., 2002). Unfortunately, the smoothing tools currently available are not advanced enough to simulate the smoothing happening in the earth.

Most of the smoothing operators are represented by convolutional-type operators (e.g., three-point operators, Gauss operators). These operators are simple and robust, but less flexible because they cannot easily change the operator length to accommodate local smoothing requirements. Moreover, the computational cost depends on the operator length. These drawbacks have been eliminated by Liu (1994) who presents a velocity-smoothing technique based on damped least squares. In this technique a smooth velocity function is sought that minimises the weighted sum of (1) the deviation between the

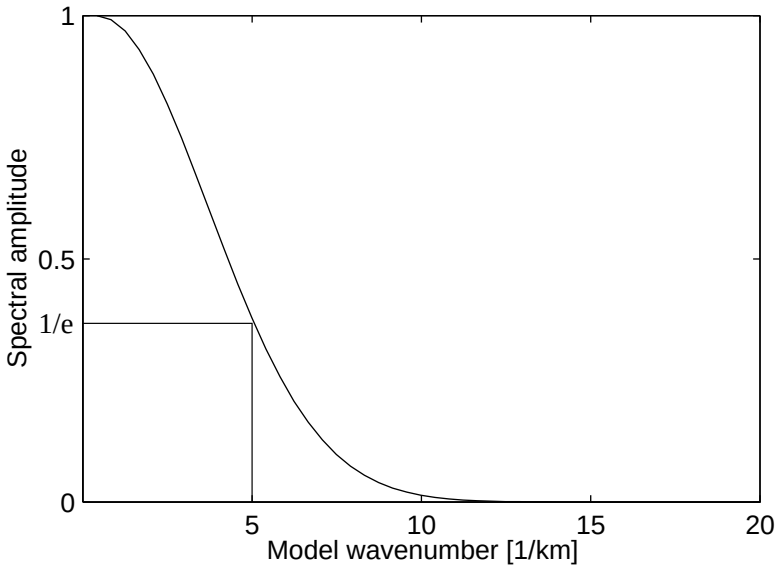

Figure 3.1: Spectral amplitude of a model with initial white spectra after multiple application of the three-point average operator. I consider the upper boundary of this filter at a wavenumber of 5 km, i.e., spatial variations below 200 m are "removed".

smooth velocity and the original one, and (2) the first derivatives of velocity. Another approach for smoothing was used by Bulant et al. (2001), who claim to construct optimally smooth models for given computational methods by minimising the Sobolev norm of the relevant derivatives of the slowness.

Smoothing of velocity models is a low pass filtering process which attenuates short period spatial velocity variations. In this thesis, I use a three-point averaging operator. The spectral amplitude of a model with initial white spectra after multiple application of the three-point average operator is shown in Figure 3.1. Because the traveltimes are proportional to the slowness, smoothing the slowness model will preserve the traveltimes better than smoothing the velocity model. To smooth the slowness model, I apply the three point average operator ($p(i) = (p(i-1) + p(i) + p(i+1))/3$) several times alternately in horizontal and vertical direction. Spatial wavelengths which are attenuated by more than 1/e are considered as removed (Figure 3.1).

The average operator applies the same weight for all three points. A comparison with an operator which uses a higher weight for the central point (e.g., $p(i) = (p(i-1) + 2p(i) + p(i+1)) * 0.25$) is given in Figure 3.2. The unsmoothed velocity model was a two layer velocity model with the initial velocity distribution of $v_1 = 2$ km/s and $v_2 = 3.5$ km/s. The interface between the layers was at 0.5 km depth. Both operators are applied 100 times. The operations required by the first operator are three additions and one division by 3, while the operations required by the second operator are two multiplications and

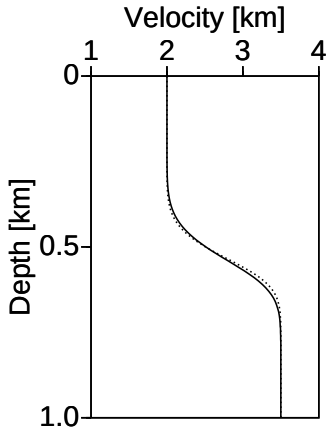

Figure 3.2: Comparison of two three-point operators. The unsmoothed velocity model has two layers (v1=2 km, v2=3.5 km). The bold line is the vertical velocity profile obtained by applying the operator with the weights (1/3, 1/3, 1/3) for 100 times. The dotted line is obtained by applying the operator with the weights (1/4, 1/2, 1/4) for 100 times.

three additions. One can also observe that the smoothing effect of the operator with equal weights is stronger than the smoothing effect of the other operator. In conclusion, the operator with equal weights has a small advantage in terms of computational time.

Smoothing the velocity model affects the computed traveltimes. I illustrate this effect in the above mentioned velocity model. The source is located at the coordinates $z = 0$ km and $x = 0.5$ km (Figure 3.3b). The dotted line in Figure 3.3a shows the vertical velocity profile which is obtained after applying 100 times the three-point averaging operator. The distance between gridpoints is 10 m. The discontinuous change in velocity at a depth of 0.5 km is replaced by a smooth change between 0.40 km and 0.65 km.

The traveltimes in the upper unsmoothed velocity layer have been computed analytically. An analytical computation was not possible in the lower layer, but two-point ray tracing provides accurate traveltimes. The traveltimes in the smoothed velocity model have been computed with the WRT technique in a high accuracy mode. The WRT technique is described in Chapter 6.

The background shading in Figure 3.3b shows the distribution of the traveltime errors. The largest errors (up to 10 ms) occur in the region close to the interface ($0.45 - 0.55$ km), but the errors decrease rapidly below this region. Below the interface region, we have small errors and a very good agreement for vertical traveltimes.

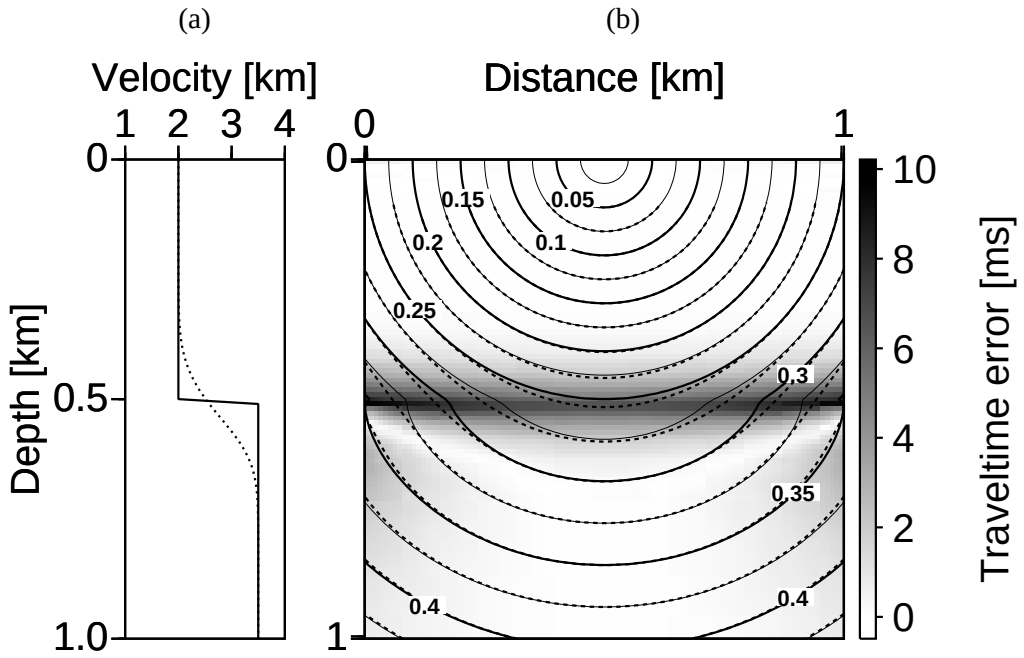

Figure 3.3: (a) The solid line shows the vertical velocity profile in the unsmoothed two-layer velocity model. The dotted line shows the corresponding smoothed profile. (b) The solid curves show wavefronts in the unsmoothed velocity model. The dashed curves show the corresponding wavefronts in the smoothed velocity model. The background shading shows the difference between the traveltimes computed in the smoothed and unsmoothed velocity model. The maximum difference is about 10 ms and occurs in the region around the interface.

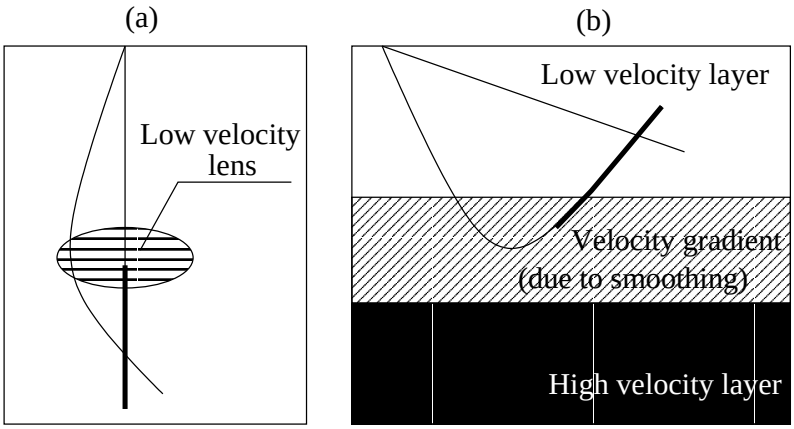

Figure 3.4: Later arrivals. The bold lines show the regions where the ray belongs to later arrivals. We distinguish between two types of later arrivals: (a) later arrivals due to low velocity regions, and (b) later arrivals due to strong velocity gradients. These gradients lead to turning rays.

It should be noted that these errors are due to different velocity models and not errors in computation. The wavefronts in the initial velocity model are broken at the interface, while the wavefronts in the smoothed velocity model are much smoother. Real wavefronts, which result from the propagation of waves with finite frequencies, are smooth; therefore, a moderate degree of smoothing leads to more realistic wavefronts.

Smoothing transforms a sharp velocity contrast in a velocity gradient. A velocity gradient between a low velocity layer and a high velocity layer may lead to diving/turning rays. In most cases turning rays are not needed in Kirchhoff migration. Avoiding their computation might considerably increase the computational speed.

Turning rays are defined by their negative value of the vertical slowness component. The ray path of such a later arrival is longer than the ray path of the corresponding first arrival (Figure 3.4b). In contrast to these later arrivals, there are other types of later arrivals which are connected to low velocity regions. A low velocity later arrival has a shorter ray path and usually a larger amplitude than the corresponding first arrival (Figure 3.4a).

The degree of smoothing can be described by the shortest remaining spatial wavelength of the velocity distribution. This wavelength is related the distance between velocity gridpoints and the integration step of the ray-tracing procedure (Section 3.5). A higher degree of smoothing allows a representation of the velocity model by a lower number of gridpoints, i.e., less computing memory is needed. Reduced memory requirement is important in 3D models which use linear interpolation for the estimation of velocities at arbitrary points (Section 3.4). The use of linear interpolation and of large integration steps increases the computational speed of ray tracing.

Smoothing the velocity model also leads to a smoother wavefront. In case of a multifolded wavefront, the spatial distribution of triplications may be reduced and the number of later arrivals may decrease. Because the number of arrivals decreases with the increase of the degree of smoothing, the computation time also decreases.

It is generally accepted that moderate smoothing has only a small influence on the image quality after migration (e.g., Versteeg, 1993; Clar et al., 1996; Operto et al., 2000). However, it is difficult to define the optimal degree of smoothing because the migration results depend also on other factors (e.g., complexity of the velocity distribution, migration method/implementation, traveltime computation method, velocity errors). There are also different smoothing techniques which have been used and different definitions for the degree of smoothing. To compare migrated images from different publications with the aim to determine the optimal degree of smoothing is difficult.

For FD migration, Versteeg (1993) shows that smooth models allow seismic imaging by prestack depth migration of very complex geological structures. However, progressive smoothing of the true velocity model gives rise to progressive degradation of the image. For the Marmousi model, Versteeg obtained optimal results by removing spatial wavelengths smaller than 200 m (200 m macromodel).

Clar et al. (1996) showed that, by using the same degree of smoothing, the images created with FD migration are better than those by Kirchhoff migration, i.e., Kirchhoff migration requires a less smooth velocity model than FD migration. Their implementation of the Kirchhoff migration uses only one arrival: the maximum energy arrival or the minimum traveltime arrival. The image created by the Kirchhoff migration which uses the maximum energy arrival was better than the image obtained from the Kirchhoff migration which uses the first arrival. Operto et al. (2000) showed that the migration image from a Kirchhoff migration which uses several arrivals is superior to the image obtained from a Kirchhoff migration which uses only one arrival.

Operto et al. (2000) also compare the migrated images obtained from two smoothed versions of the Marmousi velocity model. For the first macromodel, they remove spatial wavelengths smaller than 76 m; for the second, they remove spatial wavelengths smaller than 200 m. The best image was obtained by the Kirchhoff migration which used all arrivals computed in the less smoothed macromodel. If Kirchhoff migration uses only the first arrivals, then the quality of the migrated image in the 76 m macromodel and in the 200 m macromodel are comparable. On the one hand, the excessive smoothness of the macromodel degrades spatial resolution and the positioning of reflectors. On the other hand, the migrated image obtained for the 200 m macromodel shows larger amplitudes than the migrated image obtained for the 76 m macromodel. A smoother velocity macromodel leads to a less folded wavefront, and the energy of later arrivals “is transferred” to the first arrival. Operto et al. (2000) conclude that, when using only single arrivals, which has many advantages in terms of CPU efficiency, a compromise may be achieved by using a smoother macromodel.

3.4 Interpolation of input quantities at arbitrary points

In this section, I discuss the estimation of input quantities at arbitrary points. The input quantities for KRT are the velocity and the first-order derivatives of velocity, while for DRT the second-order derivatives of velocities are additionally needed (e.g., Červený, 2001). The integration of kinematic and dynamic ray-tracing systems along the rays requires the estimation of input quantities at arbitrary points.

For the estimation of input quantities, Ettrich and Gajewski (1996) used a linear interpolation with precomputed velocity derivatives at gridpoints. Linear interpolation is fast because the derivatives are computed once and used many times, the interpolation uses few supporting points (4 in 2D and 8 in 3D), and it uses the same interpolation coefficients for all quantities.

The accuracy of linear interpolation is sufficient if a number of 10-12 gridpoints per shortest spatial wavelength is used (Ettrich and Gajewski, 1996). This high sampling is a severe drawback for DRT, which needs 10 input quantities (one velocity, three first-order derivatives and six second-order derivatives). Using linear interpolation the memory requirement per input quantity for a $4\,\mathrm{km} \times 4\,\mathrm{km} \times 4\,\mathrm{km}$ model with the shortest spatial wavelength of 100 m is about 270 MB. The total amount of memory required for DRT is about 2.7 GB. For an efficient computation, all input quantities should be stored simultaneously in the memory (swapping the data between the memory and the hard disk decreases the computational speed considerably).

The memory requirement could be decreased by using a smoother model. However, it must be taken into account that this may affect the quality of the migrated image (see Section 3.3). There are also two other possibilities to decrease the memory requirement. The first is to renounce precomputed second-order derivatives. In this case the main reason for using linear interpolation will be lost. The second possibility is to eliminate the need for second-order derivatives by avoiding DRT.

An alternative to linear interpolation is interpolation by cardinal splines (de Boor, 1978; Thomson and Gubbins, 1982). Cardinal splines need approximately three times fewer gridpoints in each direction and do not use precomputed velocity derivatives. Using cardinal splines, the memory requirement for the above mentioned model reduces to 10 MB.

Splines supplementary smooth the velocity model, i.e., a velocity profile computed by splines will be smoother than a velocity profile computed by linear interpolation. Moreover, splines may introduce velocity variations which are not in conformity with the initial velocity model.

The interpolation by cardinal spines is slower than the linear interpolation but it is faster than the interpolation with cubical splines. For cubical splines the value at an arbitrary point depends on all values in the model, while cardinal splines use a local representation in which the value at an arbitrary point is determined only by the value at the 64 nearest gridpoints (Thomson and Gubbins, 1982).

In conclusion, the linear interpolation is fast but needs a lot of computing memory. The amount of memory considerably increases for DRT. Linear interpolation should be used only in combination with KRT in isotropic media. Cardinal splines are a better alternative for 3D DRT and for ray tracing in anisotropic media. To use the advantages of the linear interpolation, I implemented the WFC/WRT methods only on the basis of KRT.

3.5 Numerical integration of the KRT system

In this section, I review how the KRT system is numerically solved, and how the efficiency of this computation can be improved in wavefront construction methods.

The KRT system of equations (2.20) is a system of ordinary differential equations. The numerical methods which are used for solving these equations are called integration methods. The simplest integration method is the Euler method, which uses only one step for each propagation of the ray. Because of its low accuracy, the Euler method is not recommended for solving the KRT system. Two widely used alternatives are Runge-Kutta methods and predictor-corrector methods. Runge-Kutta methods propagate the ray over a traveltime step by combining the information from several smaller Euler-style steps. Predictor-corrector methods store the solution along the ray, and use those results to extrapolate the solution one step advanced; they then correct the extrapolation using derivative information at the new point. Details about integration methods can be found in many books on numerical mathematics (e.g., Press et al., 1992).

In their WFC method, Vinje et al. (1996b) used an integration method based on a Taylor expansion. In blocky models with constant second-order velocity derivatives, this procedure is able to trace a ray quickly and accurately. Because in gridbased models the second derivatives vary, more elaborate methods for solving ordinary differential equations are necessary. The most popular method for integrating the KRT system is the fourth-order Runge-Kutta method.

The traveltime step size is important for the efficiency of the integration technique, but this importance can be reduced by using a Runge-Kutta method with adaptive stepsize. At first glance this seems to be a very efficient method because it allows to trace large steps in homogeneous parts of the model. However, it must be taken into account that the tracing of one adaptive step is more expensive than the tracing of one constant step. Moreover, methods with adaptive stepsize are not applicable in wavefront construction methods.

Wavefront construction methods require that all rays are propagated with the same traveltime step. Choosing this step as a function of the smallest spatial velocity wavelength is not efficient because in most parts of the model this step is unnecessarily small. A better alternative is to choose a larger traveltime step which will be divided into two or more

steps in complex regions. The complex regions can be easily detected during the computation by comparing the values of the first-order velocity derivatives at the endpoints of the ray segment or if the computed slowness does not satisfy the eikonal equation. The accuracy and stability of ray tracing is also increased by the normalisation of $p_i p_i$ to $1/v^2$ at any step of the numerical ray tracing (e.g., Červený, 2001).

For subsurface imaging, the required traveltime accuracy is often connected with the sampling rate of the seismograms. Therefore, traveltime errors of 2 ms are usually considered acceptable.

Chapter 4

Wavefront construction method

In this chapter, I present my implementation of the 3D WFC method. This implementation is based to the WFC methods proposed by Vinje et al. (1996a), Lucio et al. (1996), and Ettrich and Gajewski (1996). Additionally, I use a new approach for the interpolation of new rays in the model.

The main characteristics of the WFC methods are (see also Figure 4.1): (1) the propagation of the ray field with a constant traveltime step (wavefront propagation), (2) the insertion of a new ray between two adjacent rays in case of insufficient illumination, and (3) the estimation of ray quantities (e.g., traveltimes) within a ray cell onto a Cartesian grid. A cell is the part of the ray tube between the last two constructed wavefronts. In the following I use the term *cell* instead of ray cell and *node* as a shortcut for the intersection point between wavefronts and rays.

4.1 Description of the method

4.1.1 Topology and propagation of the wavefront

In this section, I describe how a 3D wavefront is represented, how it is propagated, and how the initial wavefront is started.

The wavefront is represented by a network of triangles (Vinje et al., 1996a). Figure 4.2 illustrates this network in a homogeneous velocity model. The vertices of these triangles are the intersection points between rays and the wavefront. I call these vertices nodes. Adjacent (or neighbouring) rays are the rays which belong to the same triangle.

The propagation of the wavefront is in fact the propagation of each ray with a constant traveltime step. Connecting two corresponding triangles on two consecutive wavefronts leads to a geometrical construction which is known as a 3D ray cell. In other words,

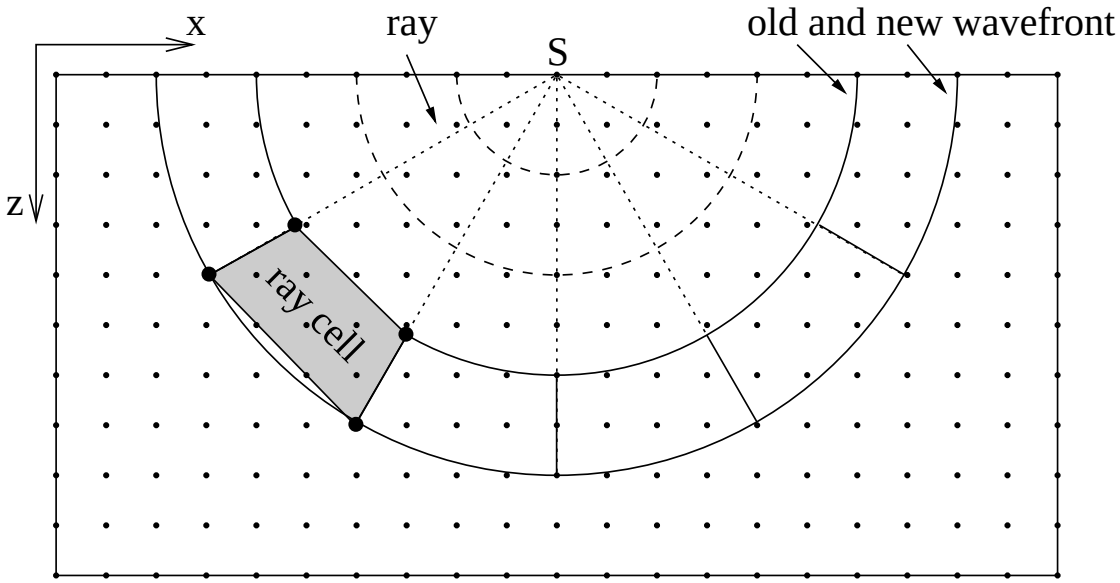

Figure 4.1: Graphical description of the WFC methods. For clarity, I illustrate the 2D version of a WFC method. The traveltimes at nodes (large dots) are computed by ray tracing. The traveltimes at gridpoints (small dots) are estimated within a ray cell. Point *S* denotes the source point.

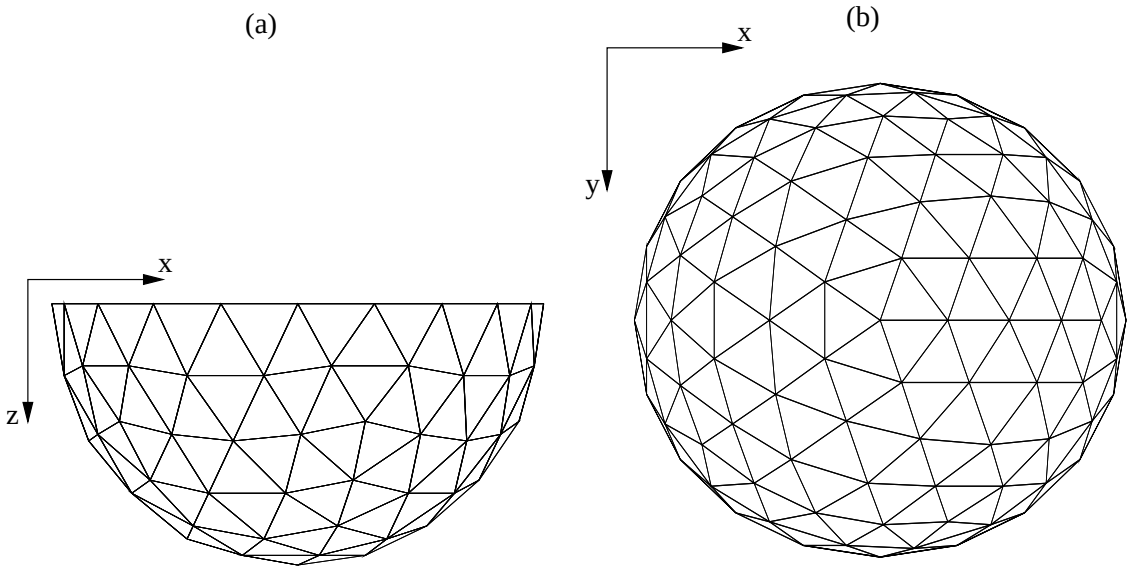

Figure 4.2: Representation of the wavefront by a triangular network. The source is at the surface of a homogeneous velocity model. (a) Viewpoint which is normal to the *xz* plane. (b) Viewpoint which is normal to the *xy* plane.

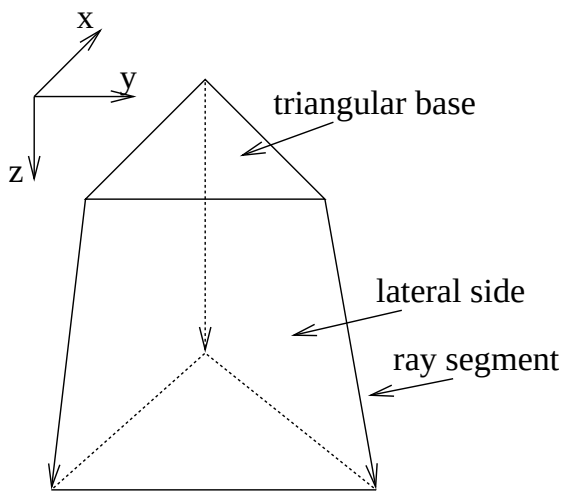

Figure 4.3: Elements of a 3D cell. The triangular bases are defined by three nodes which belong to the same wavefront. The ray segments are the approximation of rays by linear segments. The lateral sides (or cell sides) are defined by four nodes which belong to two rays. Usually, the four nodes do not belong to the same plane.

a 3D ray cell is the geometrical approximation of the region between two consecutive wavefronts and three adjacent rays. This group of three rays is also known as a ray tube. The elements of a 3D ray cell are shown in Figure 4.3. Rays, ray tubes and the sides of the tubes are enumerated (Figure 4.4).

The cell shown in Figure 4.3 has all three ray segments (resulting in six nodes) within the model. Most of the cells in the WFC method are six-node cells, but there are also other types of cells. If one or two ray segments cross the boundary of the model, or if they are completely outside of the model, then we have a "deformed" cell. This cell is partly bounded by the boundary of the model. The accuracy of traveltimes computed in deformed cells is poor because we cannot use the new wavefront for this computation. On the other hand, the accuracy of traveltimes computed in boundary regions is not essential for the quality of migrated images.

The propagation of rays is achieved by the KRT system (2.20). The KRT system is solved by a fourth-order Runge-Kutta method. I call the integration step of the Runge-Kutta method the *time step of rays*. A new wavefront is constructed from the old one by propagating the ray field with a constant traveltime step (*time step of wavefronts*). The time step of wavefronts can be the same as the time step of rays (Figure 4.5a), or can be a multiple of the time step of rays (Figure 4.5b).

After the construction of a new wavefront, traveltimes are estimated in the region between this wavefront and the previous one. Therefore, only two wavefronts are simultaneously stored in the computer memory. The traveltimes at gridpoints are estimated within cells.

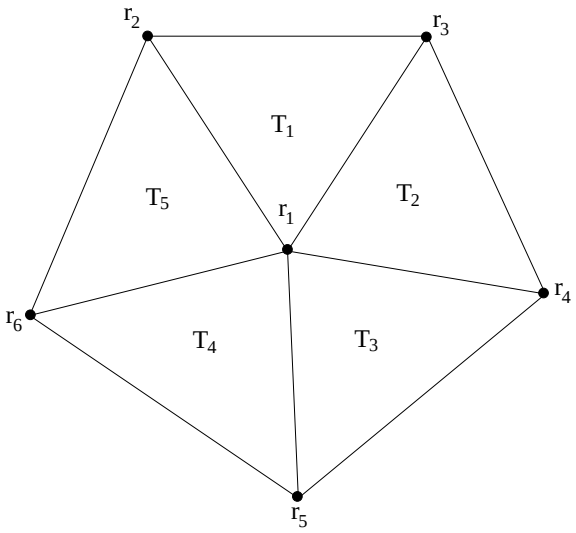

Figure 4.4: Rays, ray tubes (triangles in a wavefront section) and the sides of tubes (triangles) are enumerated. The ray number is r_i; the triangle number is T_i . The numbers satisfy the relations: $r_i < r_{i+1}$ and $T_i < T_{i+1}$, where i is an integer.

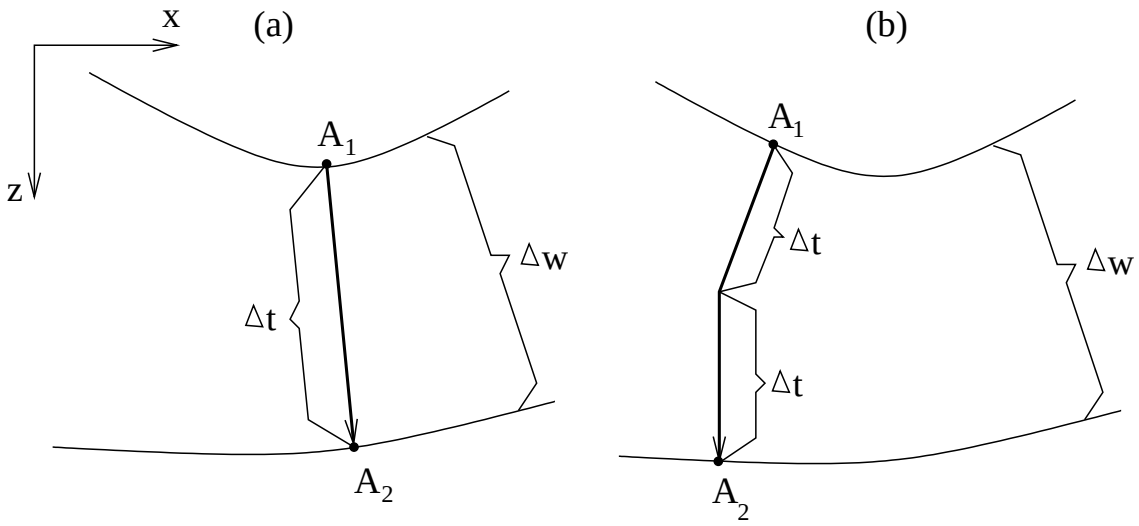

Figure 4.5: The time step of the wavefronts (Δw) and the time step of rays (Δt) can be the same (a), or they can be different (b).

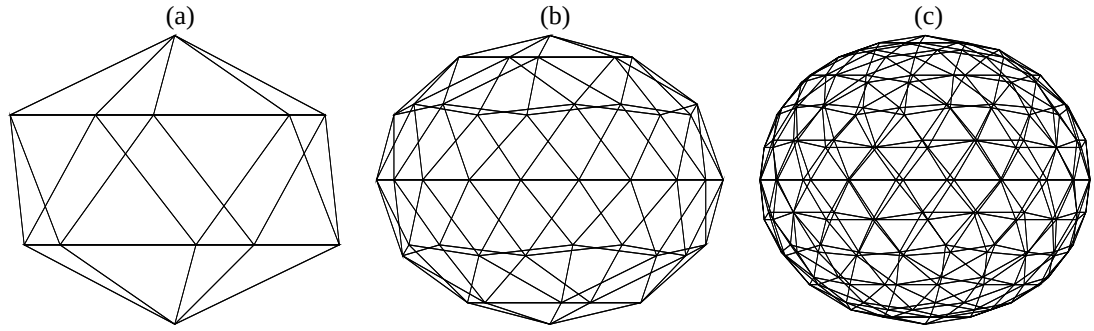

Figure 4.6: (a) The basic network of the point source is described by the icosahedron (12 nodes). (b) A polyhedron with 42 nodes is obtained by inserting a new node between two adjacent nodes. (c) Repeating this insertion procedure leads to a polyhedron with 162 rays.

To start the 3D WFC method, an initial wavefront is required. As the basis for this initial wavefront, I follow Vinje et al. (1996a) and use an icosahedron of zero radius placed at the source point. The icosahedron (Figure 4.6a) consists of twelve vertices, thirty sides and twenty regular triangles. The vertices are the intersection points between rays and wavefronts. The angle between neighbouring rays in the initial network is 63.43 degrees. In order to start with a larger number of rays, a new ray is inserted between each pair of adjacent rays. This process can be carried out iteratively. Usually, four iterations are performed, which leads to a number of 2562 starting rays.

4.1.2 Insertion of new rays

To retain a sufficient illumination by rays along the wavefronts, a new ray is inserted if one of the following insertion criteria is satisfied:

1. The distance between the adjacent nodes exceeds a predefined threshold.
2. The difference in direction between the two adjacent rays exceeds a predefined angle threshold.

Usually the coordinates of the new ray are interpolated directly on the wavefront. To determine these coordinates in a 3D WFC method, Vinje et al. (1996b) approximate the wavefront by a third-order polynomial function, Lucio et al. (1996) use a cubic Hermite interpolation, while Coman and Gajewski (2000b) use a circular approximation of the wavefront. For this implementation of the WFC method, I have developed a three-step approach to insert a new ray (Figure 4.7):

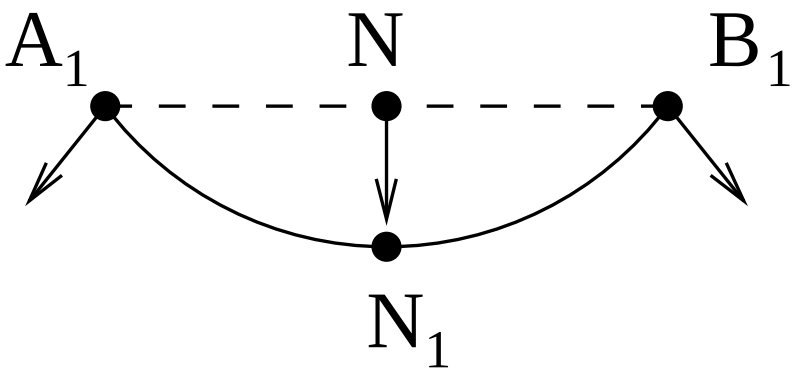

Figure 4.7: Inserting a new ray in the 3D WFC method. N is the midpoint of the segment A_1B_1. Ray tracing is used to propagate the ray from N to N_1.

1. Introduce a ray at the midpoint N between the adjacent rays. Linear interpolation is used for the slowness components. The length of the slowness vector is normalised to the inverse velocity.

2. Estimate the traveltime at point N. I apply a second-order Taylor expansion from both adjacent nodes A_1 and B_1, and store the average traveltime. The second-order traveltime derivatives are determined by finite differences of the slownesses of the adjacent rays.

3. Propagate the ray to the actual wavefront (from N to N_1) by a fourth-order Runge-Kutta method.

At most three new rays might be inserted within a ray tube (or triangular base). Inserting new rays within a ray tube leads to new ray tubes. The old tube is deleted and one of the new tubes inherits the number of the old tube. The other new tubes receive new numbers. An old tube can be split into two, three or four new tubes. Let us consider that the number of the old tube which we split is T_1 and the number of the last tube is T_n. Figure 4.8 shows some examples of the enumeration of the new tubes.

4.1.3 Estimation of traveltimes within cells

In this implementation of the 3D WFC method, traveltimes are estimated at gridpoints by linear interpolation between the wavefronts. The estimation is performed only for gridpoints within the cell. To decide whether a gridpoint is within a cell, I decompose the cell into three tetrahedra and test whether the gridpoint is within one of these tetrahedra. This approach is similar to the one proposed by Lucio et al. (1996). There are several possibilities for this decomposition, but not every decomposition ensures the contact with

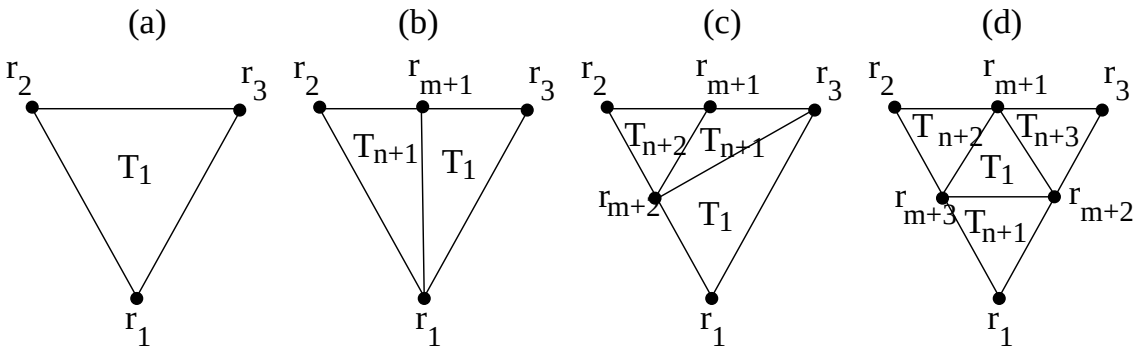

Figure 4.8: Splitting the old tube (a) into two new tubes (b), into three new tubes (c), or into four new tubes (d). One of the new tubes inherits the number of the old tube. The other new tubes receive new numbers.

tetrahedra of adjoining cells. For a successful decomposition, one has to sort the rays within a cell and follow the same decomposition rules in each cell. Lucio et al. (1996) sort the rays by their take-off angles, while I sort the rays by their number (Figure 4.9). Because a ray is defined by a single number, but by two take-off angles, sorting the rays by their number is faster.

Let us consider the cell described by the nodes $A_1, B_1, C_1, A_2, B_2, C_2$ (Figure 4.9). The relation between the ray numbers is $a < b < c$. The wavefront propagates from $A_1B_1C_1$ to $A_2B_2C_2$. I use the following rule for the decomposition of such a cell into three tetrahedra:

- The first tetrahedron is defined by nodes A_1, B_1, C_1, C_2
- The second tetrahedron is defined by nodes A_1, A_2, B_2, C_2
- The third tetrahedron is defined by nodes A_1, B_1, B_2, C_2

The decision whether a gridpoint is within a cell is replaced by the decision whether this gridpoint is within one of the three tetrahedra. The assignation of a gridpoint to a tetrahedron is well-defined.

Let us consider a gridpoint P and a tetrahedron $ABCD$ (Figure 4.10). To each side of this tetrahedron, I assign a plane which splits the space into two half-spaces. I define the positive half-space, the half-space which contains the edge of the tetrahedron outside the considered side.

The decision whether a gridpoint is within the positive half-space is simple. Let us consider three points A, B and C which define a plane Π (Figure 4.10). The equation of the

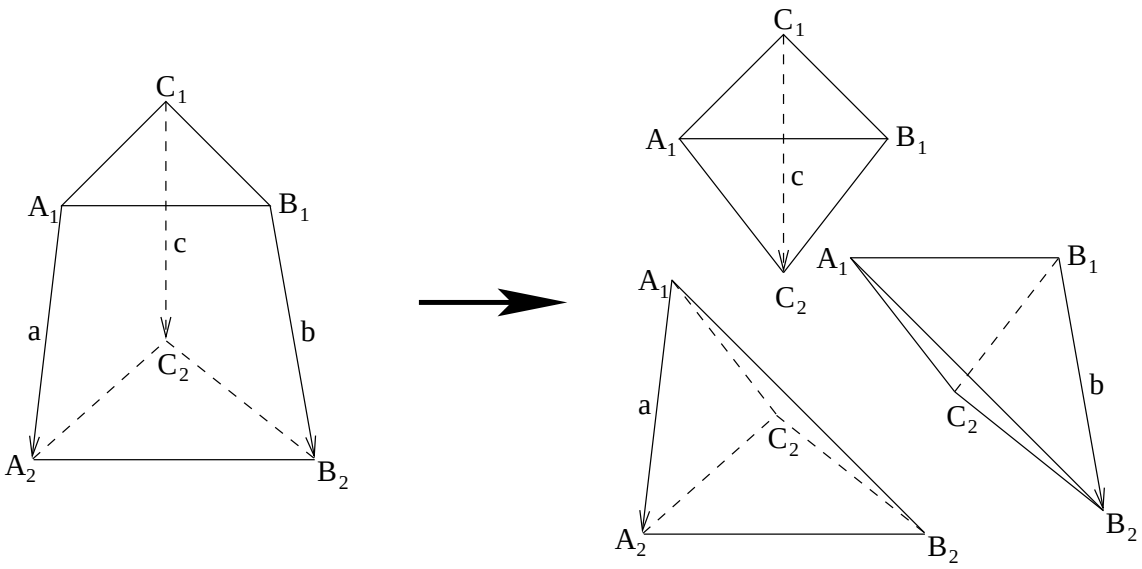

Figure 4.9: Decomposition of a cell into three tetrahedra. The relation between the numbers of the rays is $a < b < c$. The decomposition of the cell into the tetrahedra follows the same rule in each cell.

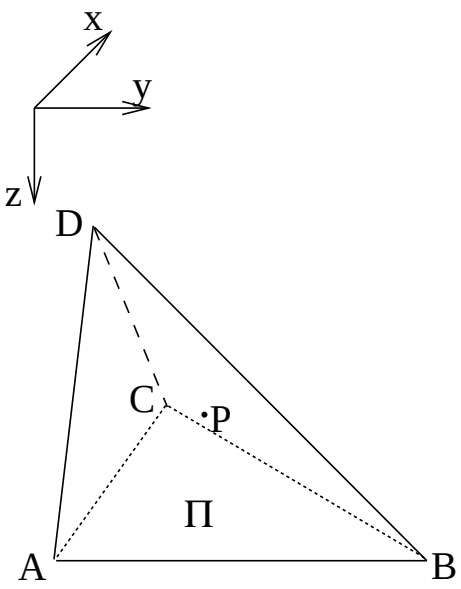

Figure 4.10: The assignation of gridpoint P to a tetrahedron. I use the location of gridpoint P with respect to each lateral plane (e.g., the plane ABC) to decide whether this gridpoint is within the tetrahedron.

plane defined by these three points is given by

$$\begin{vmatrix} x-x_A & y-y_A & z-z_A \\ x_B-x_A & y_B-y_A & z_B-z_A \\ x_C-x_A & y_C-y_A & z_C-z_A \end{vmatrix} = 0, \tag{4.1}$$

where x, y and z are Cartesian coordinates (e.g., x_A is the Cartesian coordinate x at node A). The above equation can be rewritten in the form

$$ax + by + cz + d = 0, \tag{4.2}$$

where a, b and c are components of a unit vector which is normal to the plane. The position of a point P relative to the plane Π is given by the sign of the expression

$$ax_P + by_P + cz_P + d, \tag{4.3}$$

while the distance of the point P to the plane Π is given by the modulus of expression (4.3). P is in the positive half-space if expression (4.3) has the same sign for point P and for the fourth point of the tetrahedron (in this case point D).

The point P is within the tetrahedron $ABCD$ if, for each plane defined by three points of the tetrahedron, the point P is in the same half-space as the fourth point of this tetrahedron.

In conclusion, to decide whether a gridpoint is within a tetrahedron, we need the equation of the four planes which bound this tetrahedron. For each plane, we insert the coordinates of the node opposite to the plane into the left side of the plane equation. The signs of this expression are stored. In a similar way, we compute the signs for a given gridpoint. Only if all four signs coincide, then the gridpoint is within the tetrahedron.

The assignation of a gridpoint to a cell is a robust procedure. The traveltime errors in regions with strong wavefront curvatures or with fast changes in ray direction are due to the linear interpolation approach which is used for the estimation of traveltimes within cells.

We do not intend to estimate traveltimes within caustic cells (the rays which bound a caustic cell have different KMAH indexes). Caustics are detected during the decomposition of the cell into tetrahedra by considering the relative position of the endpoints of the opposite ray. For example, if the nodes C_1 and C_2 show different signs with regard to the plane $A_1A_2B_2$, then we may assume that the ray C_1C_2 has crossed the surface between the rays A_1A_2 and B_1B_2. Caustics may be also detected by monitoring the changes of the direction of the normal to the triangular bases (Section 5.2).

4.2 Numerical example

To obtain the velocity model shown in Figure 4.11, I have extended the 2D unsmoothed Marmousi velocity model (Versteeg and Grau, 1991) in the third direction (x axis). The

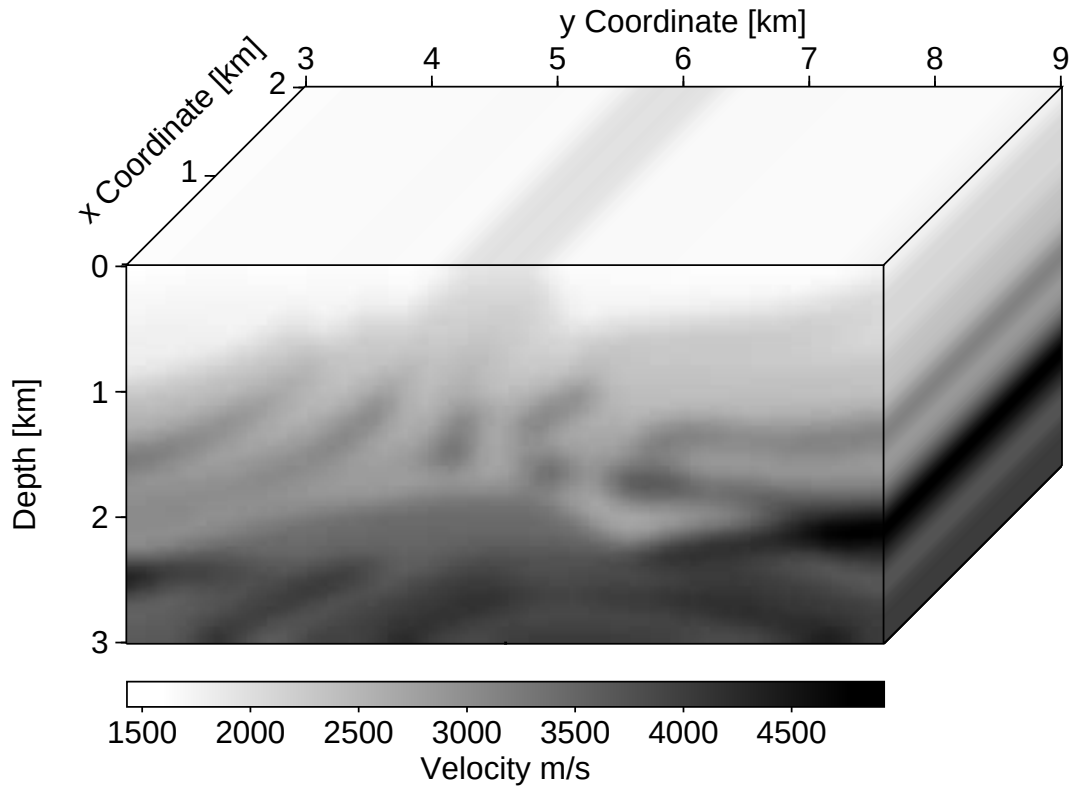

Figure 4.11: 2.5D extension of a smooth version of the 2D Marmousi velocity model. Spatial wavelengths lower than 200 m are removed

ray tracing example is carried out after smoothing this velocity model such that spatial wavelengths lower than 200 m are removed. The velocity grid was resampled from 4 m to 20 m after smoothing. The source is located at the coordinates $x = 1$ km, $y = 6$ km, and $z = 0$ km.

In this numerical example, I limit the number of computed arrivals to three. The limitation of the number of later arrivals to the first three is motivated by two facts: (1) the spatial distribution of other later arrivals is usually small, or (2) the other later arrivals are usually represented by turning rays (see Figure 3.4). Sorting the arrivals by their traveltimes, one will observe that the first two arrivals have the KMAH index 0, and in most cases the third arrival has the KMAH index 1. The KMAH index or the index of ray trajectory represents the number of caustic points along the ray (for details see Appendix C). The third arrival has the KMAH index 0 if two triplications cross each other and the second and the third arrival do not belong to the same triplication. In this numerical example, the KMAH of the third arrival was always 1. To reduce computing memory, I do not store the KMAH

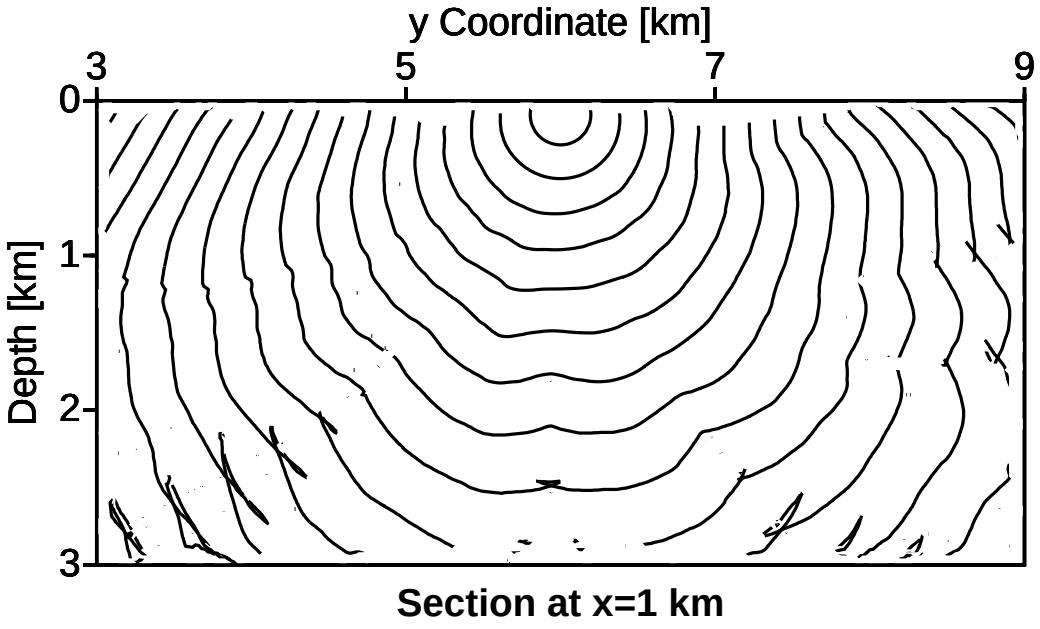

Figure 4.12: Wavefronts in the vertical section at $x = 1$ km. The distance between the wavefronts is 0.1 s.

index explicitly.

For this numerical example, a new ray is interpolated if the distance between adjacent rays was larger than 100 m or if the difference in direction between the two adjacent rays was larger than 15 degrees. For the propagation of the rays, the time step of rays was set to 10 ms and the time step of wavefronts to 40 ms. The wavefronts in the vertical section at $x = 1$ km are shown in Figure 4.12. These wavefronts are drawn on the basis of traveltimes computed by the WFC method.

The WFC method can be used alone, or in combination with a FDES. A hybrid method which combines the WFC method with a FDES is presented in the next chapter.

Chapter 5

The FDES/WFC hybrid method

5.1 Introduction

As mentioned in Section 2.3 FDESs are fast, but standard FDESs do not compute later arrival traveltimes. Several approaches have been proposed to overcome this drawback (Bevc, 1995; Benamou, 1996; Ettrich and Gajewski, 1997; Symes, 1998; Sava and Fomel, 2001).

Bevc (1995) implemented a cascaded migration scheme using the eikonal solver of van Trier and Symes (1991). The drawback of this concept is the arbitrary definition of intermediate depth levels and the access to the seismic data set for each of these depth levels.

Symes (1998) introduced an artificial horizontal interface as close to the surface as necessary to make sure that the wavefronts emitted by a surface source are single-valued at this interface. First arrival traveltimes are computed on gridpoints of this interface both from each surface source point as well as from each gridpoint below the interface. These traveltimes are denoted by $\tau(x)^{\text{down}}$ and $\tau(x)^{\text{up}}$. The traveltime $\tau(x)$ between a source point and a gridpoint below the horizontal interface is the sum of two traveltimes

$$\tau(x) = \tau^{\text{down}}(x) + \tau^{\text{up}}(x)\,, \tag{5.1}$$

if the horizontal components of the ray slowness satisfy the slowness matching condition

$$\frac{\partial \tau^{\text{up}}}{\partial x} + \frac{\partial \tau^{\text{down}}}{\partial x} = 0\,. \tag{5.2}$$

Even if the traveltimes have been assumed to be single valued in the upper and in the lower strips, there may be more than one solution of the slowness matching condition. The slowness matching condition requires at least second order accurate local solvers, which are connected with the expanding box strategy. Vidale's method (Vidale, 1988) and

its derivatives would work, but first order schemes (e.g., Podvin and Lecomte, 1991; van Trier and Symes, 1991) would result in nonconvergent global times using this approach. The drawback of this slowness matching FD method is, as in Bevc's method, the arbitrary introduction of the intermediate interface.

Benamou (1996) introduced a hybrid method which he calls big ray tracing. Rays are used to divide the model into separate regions. The traveltimes in these regions are computed by a FDES. Combining the traveltimes computed in different regions leads to multivalued traveltimes. Arbitrarily chosen rays are used for the separation of different regions. This may lead to inexact results.

Sava and Fomel (2001) proposed the Huygens wavefront tracing, which is a FD method based on the eikonal equation in the ray-coordinate system. The eikonal equation is solved by a first-order discretisation scheme, which can be interpreted in terms of the Huygens principle. The method allows the direct computation of later arrivals. The method is similar to the WFC methods in that both compute a wavefront from the preceding one. A comparison in terms of computational speed and accuracy between the two methods has been not yet performed.

For 2D media, Ettrich and Gajewski (1997) proposed a hybrid method which combines a FDES with a WFC method. The basic idea of this hybrid method is the computation of first arrivals by a FDES, the automatic detection and bounding of regions where later arrivals occur, and the final application of a WFC method to compute later arrivals. The application of a WFC method is restricted to the bounded regions.

5.2 Description of the method

In this section, I describe my implementation of the 3D FDES-WFC hybrid method.

A crucial point in the hybrid method is the detection and bounding of later arrivals. In the 2D hybrid method, Ettrich and Gajewski (1997) use the first-arrival traveltimes to detect the wavefront crossing points. This strategy is based on the fact that the beginning of a triplication corresponds to the beginning of the region with wavefront crossing points.

To detect wavefront crossing points, Ettrich and Gajewski (1997) compute the slowness vectors on the basis of first-arrival traveltimes. Points with a discontinuous slowness vector indicate regions where slower parts of the wavefront are cut off. However, in weakly smoothed media, isolated points with strong slowness vector changes which are not caused by triplications may occur. Therefore, a triplication is accepted only if three points with slowness discontinuities align. Another approach to detect wavefront crossing points was proposed by Vanelle (2002). This approach uses the sign of second order traveltime derivatives to detect the changes in wavefront curvature.

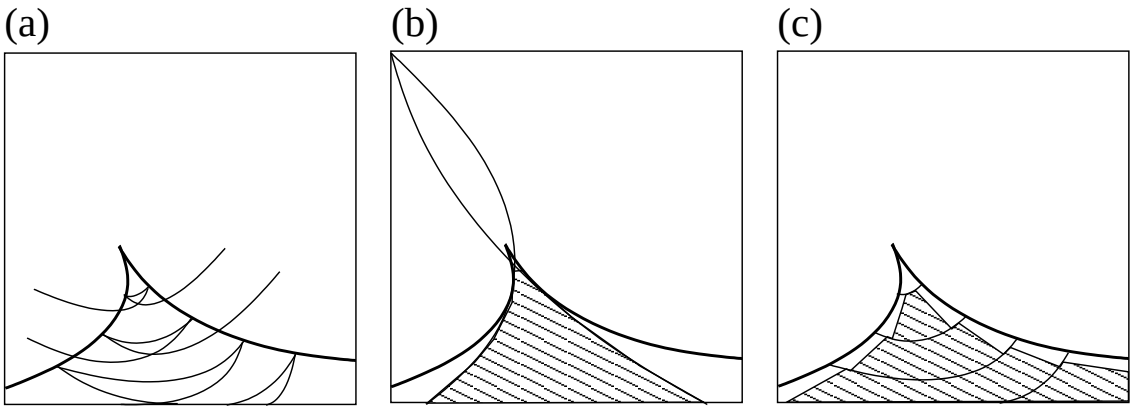

Figure 5.1: Bounding the reverse branch of a triplication. (a) Triplicated wavefronts. The caustic curve (bold curve) bounds the reverse branch of the triplication. (b) Bounding the triplication by two boundary rays. (c) Bounding the triplication by the WFC method.

To bound the reverse branch of a triplication, Ettrich and Gajewski (1997) consider the starting point of a triplication. Rays starting at two enclosing grid points are propagated backward by the method of steepest descent of traveltime. Because the take-off angles of the two rays have to be known with high accuracy a few iterations of two-point ray tracing follow. Finally, WFC is performed in the region bounded by the two rays. The algorithm for the interpolation of traveltimes on gridpoints starts after the first occurrence of a focal point.

Because the rays which bound the reverse branch of the triplication change continously, the determined bounding rays may subestimate the region with later arrivals (5.1b). Moreover, bounding of a triplication by bounding rays is difficult to implement in 3D media. Therefore, I detect and bound the reverse branch of a triplication by the WFC method. In this case the bounding rays are searched after each propagation step of the wavefront. In 2D media, the differences between these two approaches are shown in Figure 5.1.

To detect regions with folded wavefronts by the WFC method, I monitor the changes of the direction of the normal to the wavefront within a ray tube (the normal to triangular bases in Figure 4.3). The normal to the wavefront is obtained by the vectorial product $\hat{\mathbf{x}}^{(21)} \times \hat{\mathbf{x}}^{(31)}$, where the vector $\hat{\mathbf{x}}^{(21)}$ is the difference between the position vectors $\hat{\mathbf{x}}^{(2)}$ and $\hat{\mathbf{x}}^{(1)}$. Similarly, $\hat{\mathbf{x}}^{(31)} = \hat{\mathbf{x}}^{(3)} - \hat{\mathbf{x}}^{(1)}$ (Figure 5.2). The initial normal at the source point is given by the vectorial product $\hat{\mathbf{p}}_S^{(21)} \times \hat{\mathbf{p}}_S^{(31)}$ (Figure 5.2). Please note that the quantities $\hat{\mathbf{x}}^{(21)} \times \hat{\mathbf{x}}^{(31)}$and $\hat{\mathbf{p}}_S^{(21)} \times \hat{\mathbf{p}}_S^{(31)}$might also be used for the computation of the relative geometrical spreading (Section 7.2.4).

The detection of folded wavefronts by monitoring the changes of the direction of the normal to the triangular bases is shown in Figure 5.3. Figure 5.3a corresponds to an

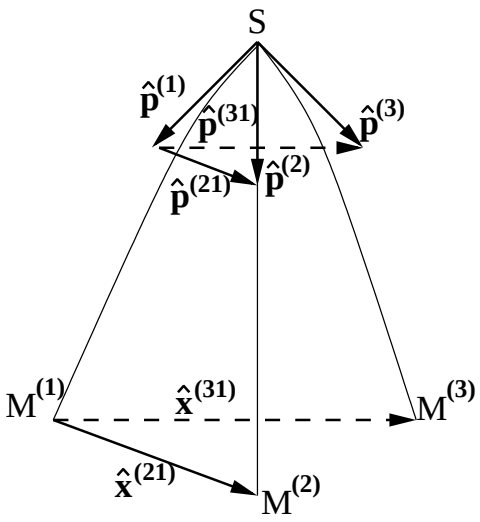

Figure 5.2: Graphical representation of the quantities which are needed for monitoring the changes of the direction of the normal to the triangular base along a ray tube. The ray tube is bounded by the rays $SM^{(l)}$; $\hat{\mathbf{x}}^{(lm)}$ is the vector between the nodes $M^{(l)}$ and $M^{(m)}$; $\hat{\mathbf{p}}_S^{(lm)}$ is the difference between $\hat{\mathbf{p}}_S^{(l)}$ and $\hat{\mathbf{p}}_S^{(m)}$; $\hat{\mathbf{p}}_S^{(l)}$ is the slowness vector at the source for the ray $SM^{(l)}$; where $l, m = 1, 2, 3$.

unfolded wavefront, Figure 5.3b,c to a folded wavefront. The wavefront in Figure 5.3b has passed a first-order caustic, while the wavefront in Figure 5.3c has passed two first-order caustics (or a second-order caustic). At a caustic point the ray tube shrinks into an arc or to a point (see Appendix C).

Let us assume that Figure 5.3a shows the situation at the old wavefront, while Figure 5.3b or Figure 5.3c shows the situations at the new wavefront. We observe that the first-order caustic (Figure 5.3b) is easily detected by the change in the direction of the normal to the triangle I. The direct detection of two consecutive first-order caustics (Figure 5.3c) is not possible because the normal to the triangle I does not change (actually it changes twice). An indirect detection is possible by considering the changes in the direction of the normal at the adjacent triangles (II, III, and IV). However, this approach is too expensive for practical applications.

5.3 Numerical example

For the below mentioned numerical examples, the first arrival traveltimes are computed by Vidale's FDES (Vidale, 1990) using an implementation of Leidenfrost (1998). The detection, bounding and computation of later arrival traveltimes was done by the WFC

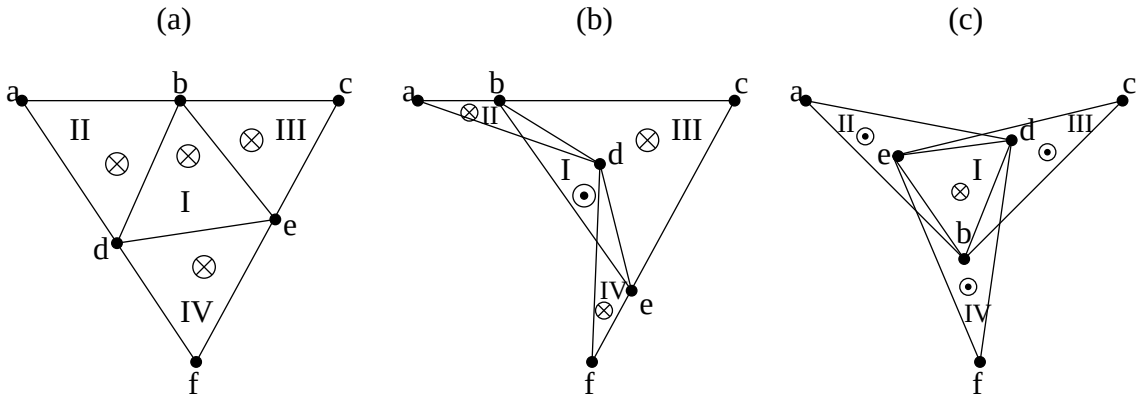

Figure 5.3: Detection of folded wavefronts by monitoring the surface normal to the wavefront. The small letters denote rays, while the Roman numbers denote triangles (ray tubes).⊗ indicates an upward surface normal, ⊙ indicates a downward surface normal (looking from the source). (a) Assumed original arrangement of three adjoining triangles. (b) The ray d intersects the surface between the rays b and e. The normal to the triangle I changes the direction. (c) Each ray within the triangle I crosses the surface between the two other adjacent rays. The triangle I changes its normal twice (i.e., new normal is the same with old normal), but the attached triangles will have reverse normal direction.

method described in Chapter 4. To compare the hybrid method with the WFC, I select for the WFC method a comparable accuracy as for the FDES. The gridpoint distance of the output grid (traveltime grid) and of the velocity grid was the same.

In this section I will apply the hybrid method to three velocity models. The first model has two layers separated by a dipping interface (Figure 5.4). The traveltime map is single-valued. In this numerical example the hybrid method was about four times faster than the WFC method alone.

The second velocity model is a 2.5D version of the Marmousi model (Figure 5.5). To obtain this velocity model, I have extended the 2D unsmoothed Marmousi velocity model in the third direction (x axis). The ray tracing example is carried out after smoothing this velocity model such that spatial wavelengths lower than 120 m are removed. The velocity grid was resampled from 4 m to 12.5 m after smoothing. The source is located at the coordinates $x = 612.5$ m, $y = 6000$ m and $z = 62.5$ m.

The wavefronts from first-arrival traveltimes for the vertical section at $x = 612.5$ m are displayed in Figure 5.6. The regions with later-arrival traveltimes are displayed in Figure 5.7. In this example, the hybrid method computes the traveltimes two times faster than the WFC method alone.

Figure 5.8 shows the qualitative variation of the CPU time as a function of the number of gridpoints with later arrival. The hybrid method is faster than the WFC method, and the

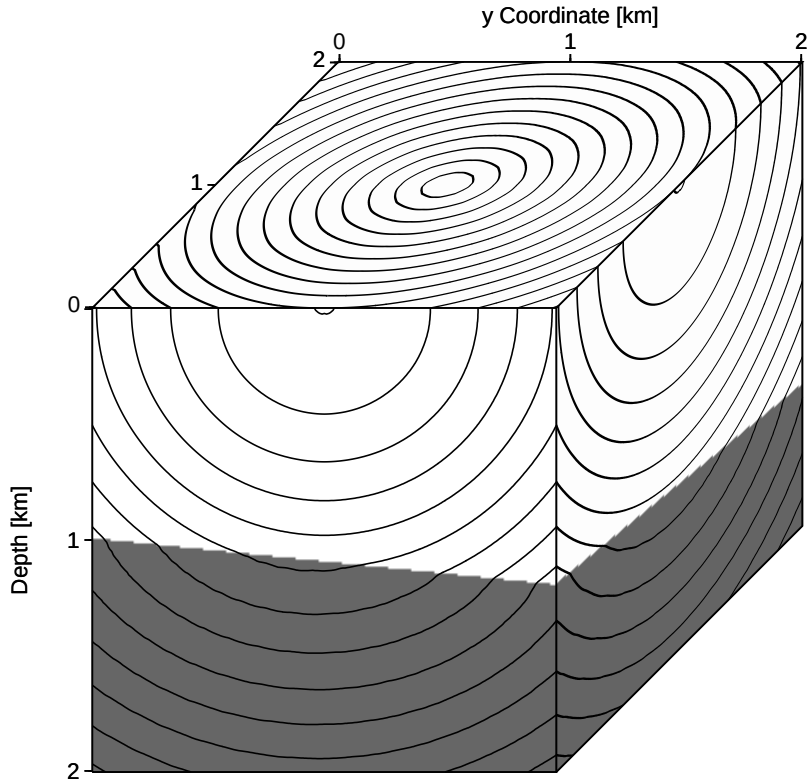

Figure 5.4: Two-layer model and isochrones. The upper layer has a velocity of 2 km/s. The layer below has a velocity of 2.5 km/s. The model has 201 gridpoints in each direction. The distance between gridpoints is 10 m. The source position is at 1 km, 1 km and 0.01 km.

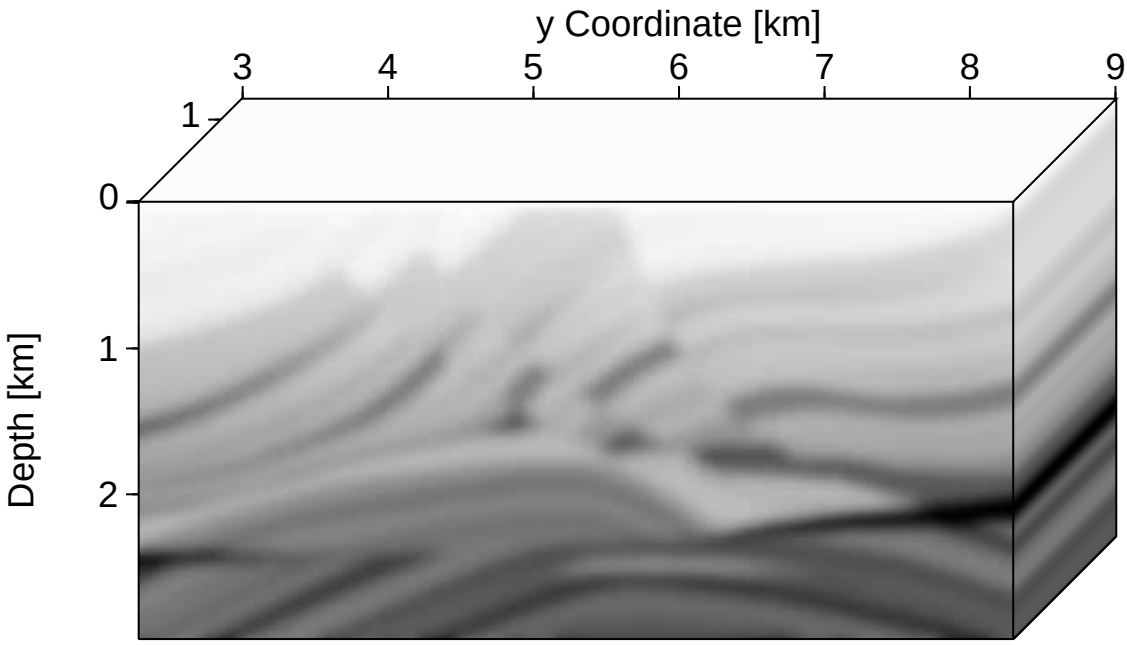

Figure 5.5: 2.5D extension of a smooth version of the 2D Marmousi velocity model. Spatial wavelengths lower than 120 m are removed.

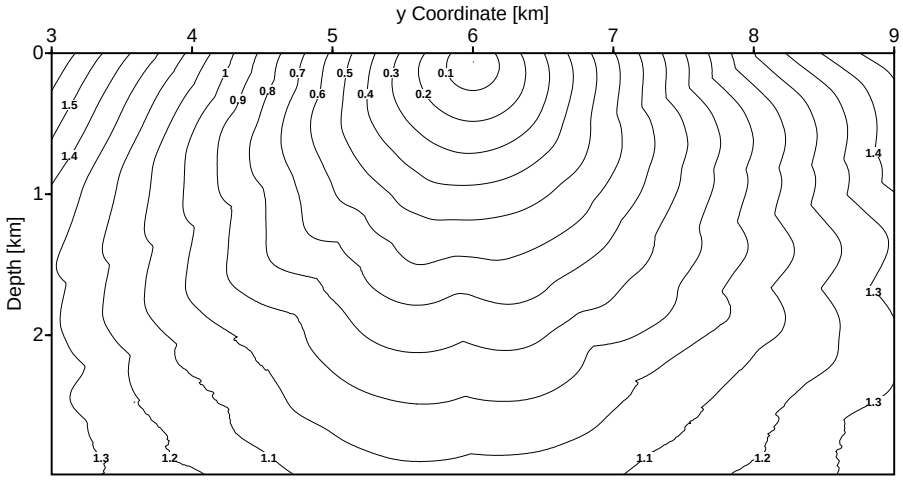

Figure 5.6: Wavefronts (only first arrivals) in the vertical section at $x = 612.5$ m. The rough wavefronts in the lower left and lower right corner of the model are due to the poor accuracy of the FDES solver. The poor accuracy is connected with a grid size that is too large for the existing velocity contrast.

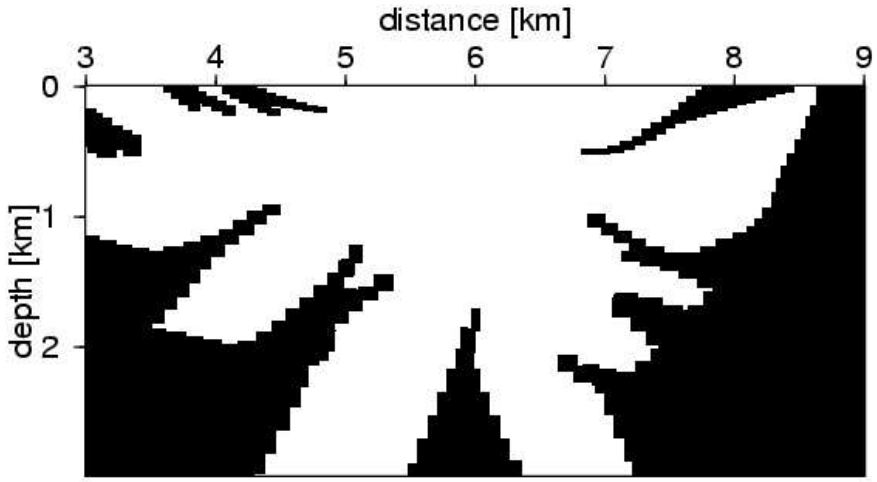

Figure 5.7: The black regions show domains where later-arrival traveltimes were computed.

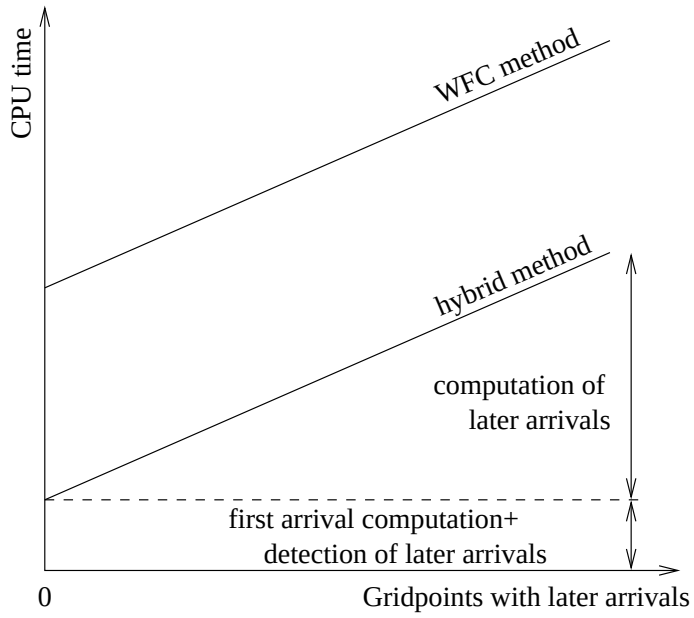

Figure 5.8: CPU time for the hybrid method and the WFC method. The CPU time for the hybrid method is the sum of the CPU time for computing first arrivals, for detecting later arrivals and for computing later arrivals. The CPU time difference between the WFC method and the hybrid method is not a function of the number of gridpoints with later arrivals.

CPU time difference between both methods is not a function of the number of gridpoints with later arrivals (the computation of later arrivals is the same in both methods).

The third velocity model is shown in Figure 4.11. For the WFC method, I use the same input parameters as in Section 4.2: a new ray is interpolated if the distance between adjacent rays was larger than 100 m or if the difference in direction between the two adjacent rays was larger than 15 degrees; the rays are propagated with the time step of rays to 10 ms; the time step of wavefronts was 40 ms.

The wavefronts in the vertical section at $x = 1$ km are shown in Figure 5.9. These wavefronts are drawn on the basis of first arrivals computed by the FDES, and of third arrivals computed by the WFC method.

Comparing Figure 4.12 with Figure 5.9, we observe that second arrivals are not computed by the hybrid method, and that the wavefronts of the first-arrivals in the lower left corner of the model ($z > 2.5$ km, $y < 3.5$ km) differ. These differences are due to the poor accuracy of the FDES. The poor accuracy is connected with a grid size that is too large for the existing velocity contrasts. The accuracy of the FDES can be increased by using a smaller velocity grid or by using a higher degree of smoothing of the velocity model.

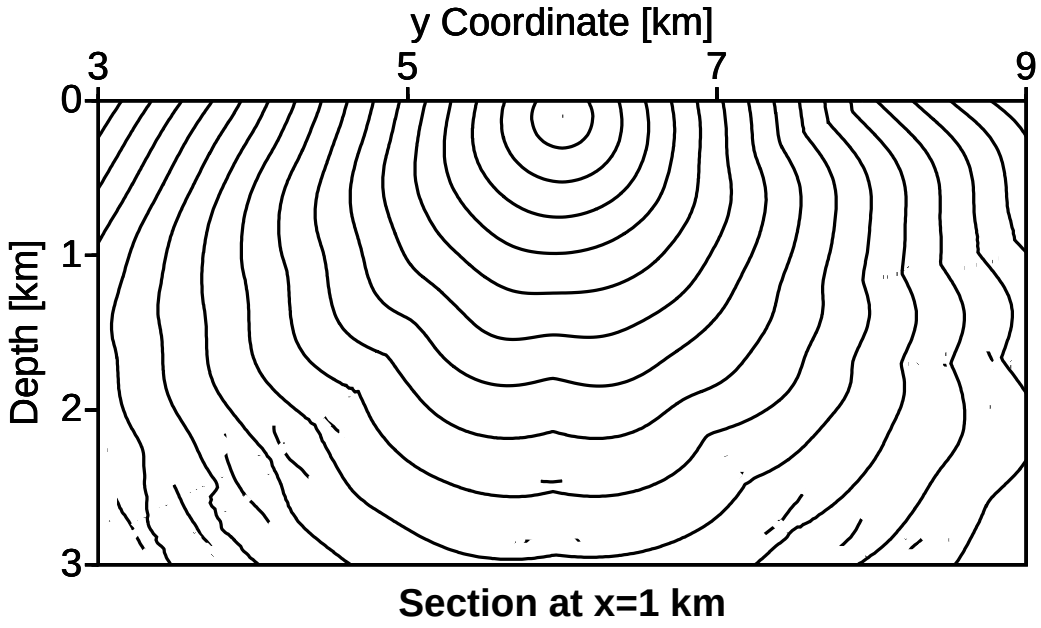

Figure 5.9: Wavefronts in the vertical section at $x = 1$ km. The distance between the wavefronts is 0.1 s.

5.4 Discussions

The 3D FDES-WFC hybrid method consists of three steps:

1. Computation of first-arrival traveltimes with a FDES
2. Detection of folded wavefronts with the WFC method
3. Computation of later-arrival traveltimes with the WFC method

In the numerical examples presented above the hybrid method was between two and four times faster than the WFC method alone. Because the FDES does not allow a flexible choice of the output grid (traveltime grid) with respect to the grid distance and target region, the output grid was the same as the velocity grid. This output grid is also a coarse grid, i.e., the traveltimes on the migration grid are obtained during the migration by interpolation (Vanelle and Gajewski, 2002). However, this coarse grid is finer than the coarse grid that may be used by the WFC method alone.

In the above mentioned comparisons, the WFC method computes the traveltimes on the same grid as the hybrid method, i.e., the output grid was not optimal for the WFC method. For a ratio of 5 between the grid distance of the output grid to the grid distance of the velocity grid, the WFC method computes 125 (5^3) times fewer traveltimes. In such situation, the WFC method may be faster than the hybrid method.

The hybrid method does not compute the arrivals between the wavefront crossing point and the focal point ("the second arrivals"). Because of this limitation, the hybrid method is not recommended in cases where the computation of all arrivals is important.

Chapter 6

2D wavefront-oriented ray tracing

6.1 Introduction

The WRT technique is related to the WFC method described in Chapter 4, but there are several important differences between these two methods.

The insertion of a new ray by interpolation on the wavefront is an important feature of the WFC methods (e.g., Vinje et al., 1993). However, in the WRT technique, I prefer to insert a new ray by tracing it from the source because of the higher accuracy. This insertion approach also allows a lower overall ray density. The insertion of a new ray by tracing it from the source was also used in the controlled initial-value ray-tracing algorithm by Bulant (1999). In this algorithm, the wave field is estimated only at the final surface (e.g., a horizontal plane in the target area) and the decomposition into cells is completely separated from the interpolation of traveltimes. Evaluating the ray field only at the final surface, however, leads to oversampling by cells (e.g., near the source region) and the advantages of the WFC methods are only partly used.

Usually, two main classes of criteria have been used to decide whether to insert a new ray. The first class (e.g., Sun, 1992) uses the distance between two adjacent nodes and the difference in direction between adjacent rays, while the second class (e.g., Lambaré et al., 1996) uses the error of the paraxial approximation in terms of coordinates and slownesses. The first class of criteria leads to undersampling in caustic regions (Lambaré et al., 1996), while the second one might overlook small-scale velocity anomalies. None of the above mentioned criteria control the errors connected with the interpolation of new rays on the wavefront. To reduce these errors, Vinje (1997) suggested an insertion criterion based on probe rays.

I insert a new ray if one of the following criteria is satisfied: (1) the distance between the adjacent rays (at nodes) is larger than a predefined threshold (upper distance threshold),

(2) the difference in wavefront curvature between the adjacent rays is larger than a predefined threshold and the distance between the adjacent rays is larger than a predefined lower distance threshold, (3) the adjacent rays cross each other and the distance between the adjacent rays is larger than the lower distance threshold. The first criterion avoids undersampling of small-scale velocity anomalies. The second criterion controls the error of the traveltime estimation within the cells, and together with the third criterion avoids undersampling in caustic zones. On the other hand, oversampling in caustic regions is avoided by the lower distance threshold. The above mentioned criteria allow to control the accuracy of the WRT technique.

For the estimation of ray quantities (e.g., traveltimes, slownesses, amplitudes) within cells, Vinje et al. (1993) projected the gridpoint on the old wavefront (old wavefront, see Figure 4.1), interpolated the ray quantities at the projection point and traced a ray back to the gridpoint. Lambaré et al. (1996) split the cell into two triangles and performed linear interpolation within triangles, while Bulant and Klimeš (1999) proposed a bicubic interpolation of traveltimes. Here, I propose a distance-weighted averaging of extrapolated traveltimes. The extrapolation is performed from nodes to gridpoints where the wavefront curvature is taken into account.

The efficiency of the WRT technique strongly depends on the input parameters which control the wavefront and ray densities. On the basis of traveltimes computed in a smoothed Marmousi model, I analyse these dependencies and suggest rules for an optimal choice of input parameters.

6.2 Description of the method

6.2.1 Detection of caustics

To know the number of caustics passed by each ray (KMAH index) is important because:

- It allows the detection of cells bounded by rays which have passed a different number of caustics. I call these cells caustic cells. The region with caustic cells and in the close vicinity of caustic cells, I call caustic region.
- It avoids the estimation of traveltimes within caustic cells.
- It allows a user defined accuracy for the localisation of caustics by a controlled increase of the ray density in caustic cells. Ideally, the caustic cells should be very small.

The number of caustics along each ray can be computed by DRT. However, because of efficiency reasons, I decided to implement the WRT technique using only KRT (see Section 3.4).

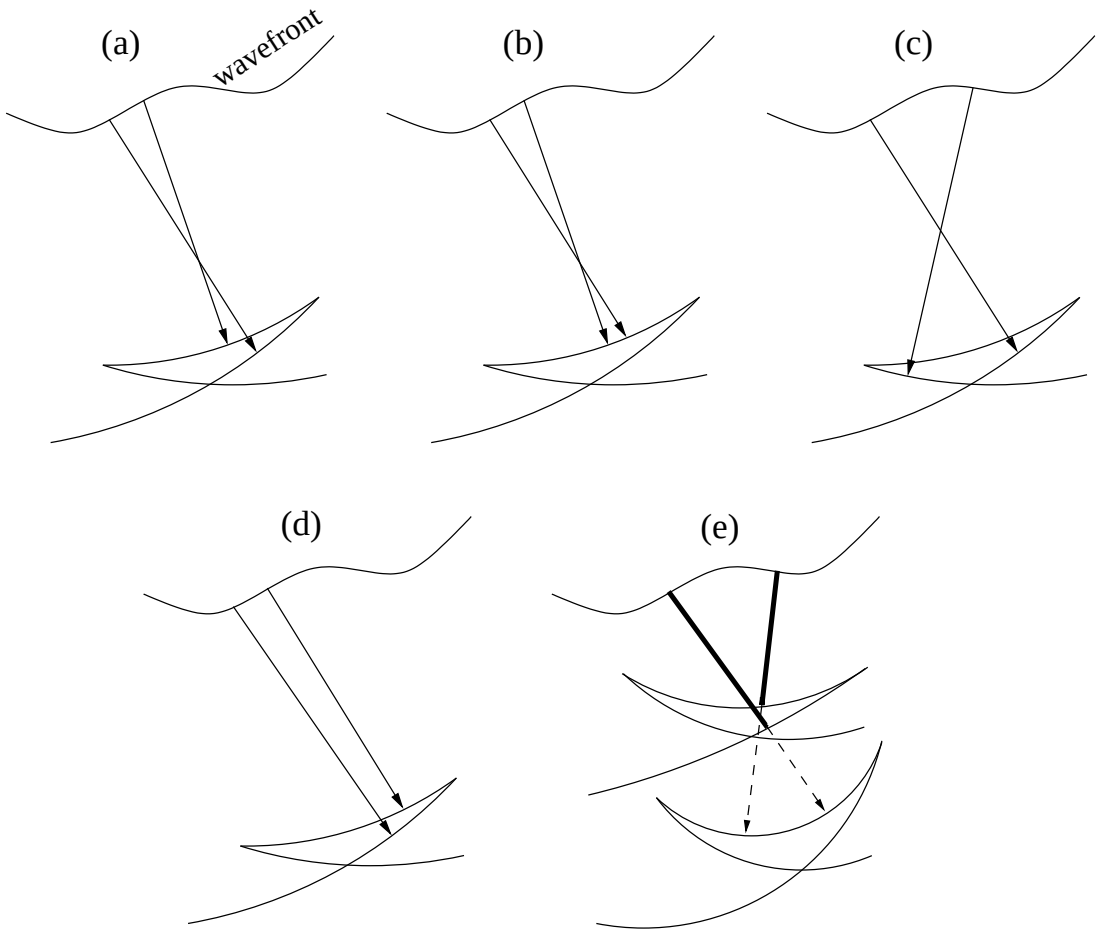

Figure 6.1: Relation between caustics and two adjacent rays which cross each other. For simplicity, the rays are represented as straight lines. (Real rays are curved and normal to the wavefronts.) (a) Only one ray passed the caustic. (b) Both rays passed the caustic. (c) No ray passed the caustic. (d) One ray passed the caustic but the adjacent ray did not cross each other. (e) The bold ray segments do not cross each other, and the dashed ray segments do not cross each other.

The intersection of two adjacent ray segments indicates that the wavefront in that region is folded, but it is not possible to specify which of the two rays has passed the caustic. Usually one ray (Figure 6.1a), or both rays (Figure 6.1b) passed the caustic, but it might happen that none of them passed the caustic (Figure 6.1c). On the other hand, Figure 6.1d shows the situation where one ray passed the caustic, but it did not cross its adjacent ray yet. Moreover, Figure 6.1e shows a case where both rays passed the caustic, but we did not observe that these rays cross each other. Indeed, the bold ray segments did not cross each other, nor the dashed ray segments cross each other.

Fortunately, missing the KMAH-index does not affect the accuracy of estimated traveltimes seriously. I do not estimate traveltimes in cells where two adjacent rays cross each other. Moreover, if two adjacent rays cross each other then a new ray is inserted between them (see Section 6.2.3). Usually, the ray density in caustic regions is high. The cells are small and are localised close to the caustics.

The estimation of traveltimes in cells where the adjacent rays do not cross each other takes the wavefront curvature into account. The approximation of the wavefront curvature is described in the next section.

6.2.2 Approximation of the wavefront curvature

Figure 6.2 shows how the wavefront curvature at node A is approximated by using the slowness vector at this node and the positions of the nodes A and B. The node B is on the same wavefront. The position of the nodes and the direction of the slowness vector are sufficient to construct a circle which passes the nodes. I approximate the radius of the wavefront curvature at node A by the radius of this circle (Figure 6.2).

The centre $O(x_O, z_O)$ (x_O, z_O denote the Cartesian coordinates of point O) of the circle is given by the intersection point between two lines: the first one passes the midpoint $M(x_M, z_M)$ between the nodes $A(x_A, z_A)$ and $B(x_B, z_B)$, and is perpendicular to the segment defined by these nodes; the second one passes the node A and points into the direction of the slowness vector.

The wavefront curvature is used not only for the extrapolation of traveltimes within cells but also as an insertion criterion for new rays. The insertion approach is described in the next section.

6.2.3 Insertion of new rays

In this section, I describe the approach used for the insertion of a new ray and the criteria for the decision when to insert a new ray. The insertion criteria should optimise the illumination of the model. An optimal illumination (or optimal ray density) allows the

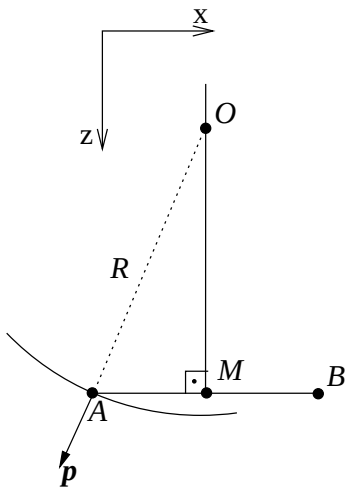

Figure 6.2: Approximation of the wavefront curvature at node A. The radius of the wavefront curvature at node A is approximated by the distance $|\overline{OA}|$. Point M is the midpoint of the segment $\overline{AB}$. Point O is the centre of the circle which passes the nodes A and B. Point O is obtained by the intersection of the continuation of the slowness vector $\mathbf{p}$ at A with the normal to the segment $\overline{AB}$ at M.

computation of sufficiently accurate traveltimes from a minimum number of rays (undersampling and oversampling are avoided).

Usually in WFC, the initial conditions at the wavefront of the inserted ray are obtained by interpolation from two adjacent rays (parent rays). Due to the interpolation involved to obtain the initial conditions at the wavefront, the accuracy of such ray is always worse than the accuracy of the parent rays. The accuracy of the interpolation strongly depends on the distance of the parent rays.

In WRT we insert a new ray by tracing it directly from the source. The initial conditions of this ray are the known source coordinates and the bisector of the angle between the parent rays at the source. These initial conditions are error free. Thus, the accuracy of the inserted rays is the same as for the parent rays. This accuracy only depends on the method used to integrate the ray tracing equations. Since the accuracy of rays inserted by tracing from the source does not depend on the distance of the parent rays, a lower ray density than for the insertion by interpolation can be used without degrading the accuracy of grid traveltimes interpolated from ray nodes. For the ray tracing we use a double-precision representation of numbers to allow small changes in slowness direction if initial conditions of parent rays vary only slightly.

A new ray between two adjacent rays is inserted if one of the following three criteria is satisfied:

1. The distance between the adjacent nodes exceeds a predefined upper distance threshold (UDT).

2. The difference in wavefront curvature between the rays exceeds a predefined threshold (time-difference threshold, see below) and the distance between the rays is larger than a predefined lower distance threshold (LDT).

3. The adjacent rays cross each other and the distance between them is larger than the LDT.

For a more descriptive view I express the difference in wavefront curvature in time units (milliseconds) by dividing the difference of the determined radii of curvature by the velocity at the midpoint between the adjacent nodes. In the following I use *time difference* as a shortcut for the difference in wavefront curvature between two adjacent rays and *node distance* for the distance between two adjacent nodes, i.e., rays.

The insertion criteria complement each other. If the insertion criteria 2 and 3 were omitted, the insertion criterion 1 alone would not lead to a sufficient illumination in caustic regions (Lambaré et al., 1996). If the LDT were omitted, the last two insertion criteria may lead to oversampling by rays in caustic regions. If the first insertion criterion were omitted, the second insertion criterion would allow the existence of very large distances between two adjacent rays and it could happen that small-scale velocity anomalies are undersampled (Fig. 6.3a). Undersampling leads to large traveltimes errors (Fig. 6.3b) below the anomalies. In the example shown, the maximum traveltime error is about 18 ms compared to reference traveltimes. The reference traveltimes at gridpoints were obtained by two-point ray tracing.

Fig. 6.3c shows the distribution of rays when all three insertion criteria are used. This situation corresponds to the procedure applied in the WRT technique. In this example, WRT starts the computation with only five rays but illuminates the velocity anomaly whereas the example using only criteria (2) and (3) misses the anomaly although much more rays were initialised at the source. The reduced number of rays in Fig. 6.3c is possible since the first insertion criterion ensures that the distance between adjacent rays stays below a specified threshold. By keeping a sufficient illumination the total number of rays in the model is reduced which positively impacts the CPU time. This countermands the additional effort to trace the rays from the source.

The second insertion criterion is related to the approach which is used for the estimation of traveltimes within cells. As we apply the wavefront curvature for this estimation, the accuracy of computed traveltimes depends on the difference in wavefront curvature between adjacent rays. If the estimation of traveltimes within cells does not apply the wavefront curvature but the ray direction, then the use of the difference in ray direction as a second insertion criterion is justified. A comparison between the two insertion criteria is shown in Fig. 6.4.

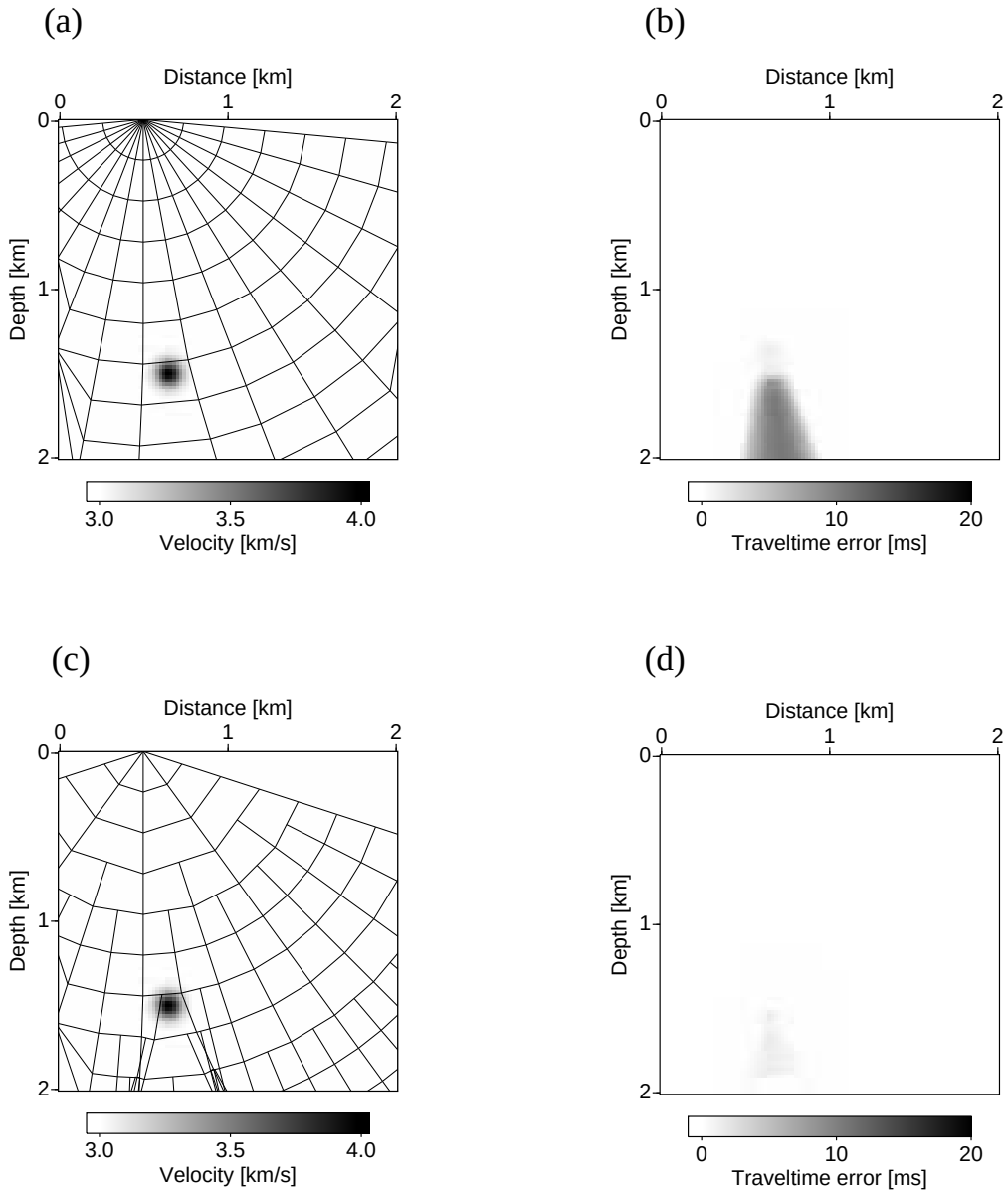

Figure 6.3: Insertion criteria. (a) Distribution of cells in the case when only the second and the third insertion criteria are used. The background shows the velocity model. The velocity of the homogeneous medium is 3 km/s, while the velocity in the centre of the lens (x=0.65 km, z=1.5 km) is 4 km/s. The diameter of the lens is 200 m. (b) Absolute traveltime error for case (a). (c) Distribution of cells in the case when all three insertion criteria are used. (All rays start from the source, but only the ray segments which build cells are drawn.) (d) Absolute traveltime error for case (c).

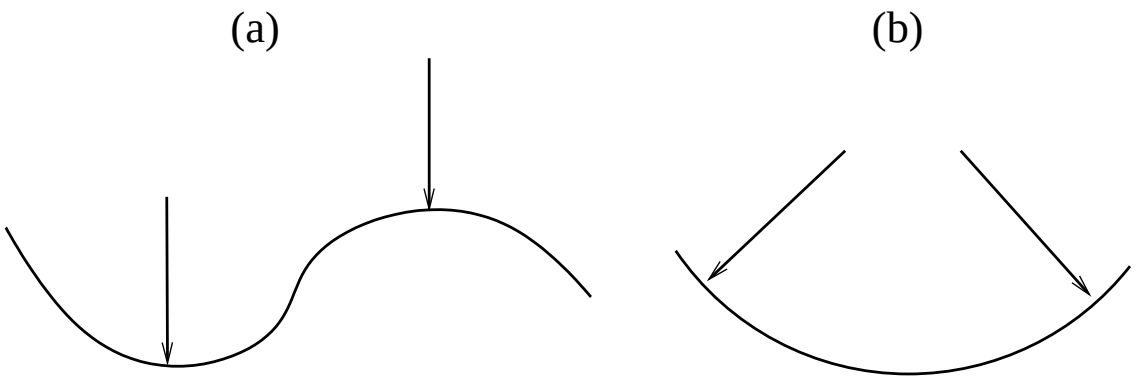

Figure 6.4: Comparison between the ray-direction insertion criterion and the wavefront-curvature insertion criterion. The ray-direction criterion will not insert a new ray in case (a) and will insert a new ray in case (b). The wavefront-curvature criterion will do the opposite.

An optimal ray density is a precondition for an accurate estimation of traveltimes within cells. This estimation is described in the next section.

6.2.4 Estimation of traveltimes within cells

The ray-tracing procedure computes the traveltimes at nodes, but for Kirchhoff depth migration the traveltimes are needed on a rectangular grid. The WRT technique uses the traveltimes to estimate the gridpoint-traveltimes.

The estimation is carried out within cells, which are usually quadrangular. Non-quadrangular cells are used close to the boundaries of the model. The boundaries of the quadrangular cells are two adjacent ray segments and two adjacent wavefront segments (Figure 6.5). Linear ray segments and wavefront segments are the local approximation of rays and wavefronts. The corners of quadrangular cells usually correspond to nodes, but a special situation occurs when a new ray is inserted. In order to ensure that the new cells are in contact with the preceding cell, the node of the inserted ray is projected onto the wavefront segment of the preceding cell (Figure 6.5). A similar approach was used by Lambaré et al. (1996).

After defining a cell, I search for the gridpoints within this cell. To limit the searching area, I bound the cell by a rectangular area (Figure 6.6) and check each gridpoint in this area for its position relative to the cell. To decide if the gridpoint is inside or outside the cell, I use the approach proposed by Ettrich and Gajewski (1996) (see Figure 6.7). They connected the gridpoint to every cell-corner by a vector and perform the vector products $\mathbf{d}_{A_1} \times \mathbf{d}_{B_1}$, $\mathbf{d}_{B_1} \times \mathbf{d}_{B_2}$, $\mathbf{d}_{B_2} \times \mathbf{d}_{A_2}$, and $\mathbf{d}_{A_2} \times \mathbf{d}_{A_1}$. The gridpoint is inside the cell if the signs of the resulting vector products are the same.

I have mentioned above that the insertion of a new ray by tracing it from the source allows

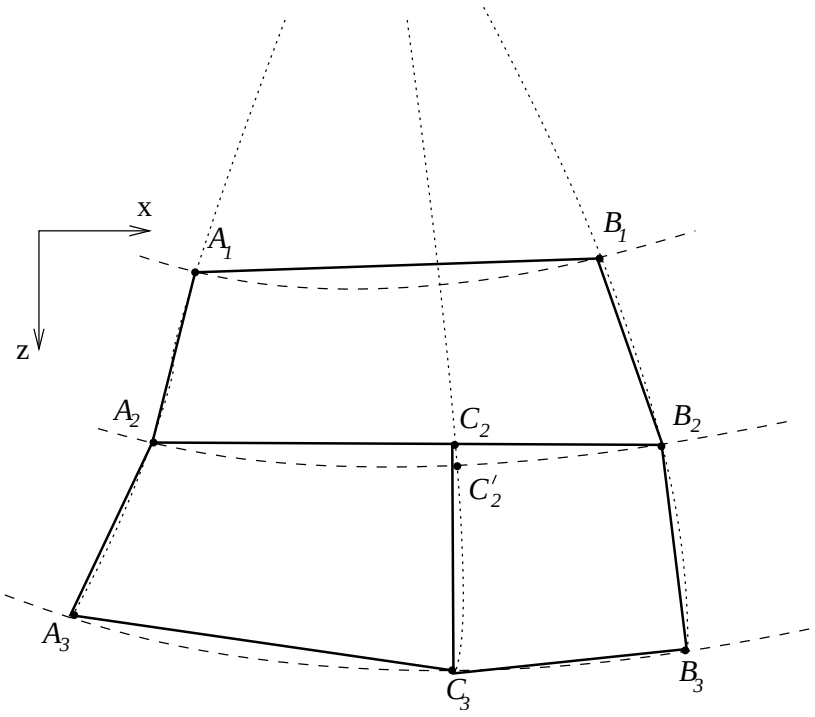

Figure 6.5: Rays, wavefronts and cells. The dotted lines are rays, the dashed lines are wavefronts, and the solid lines are linearised rays or wavefronts (ray and wavefront segments). The nodes C_2' and C_3 lie on a newly inserted ray. The quadrangles indicated by solid lines are cells. In most cases the cell corners lie on nodes. Exceptions occur when a new ray is inserted. To ensure a continuous coverage with cells the node C_2' is projected on the wavefront segment A_2B_2 ($C_2'C_2$ is perpendicular to the solid line A_2B_2).

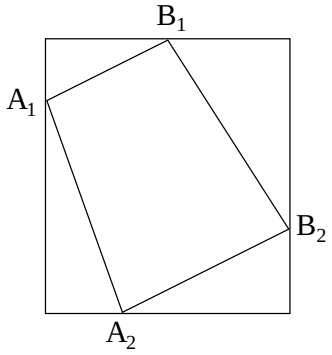

Figure 6.6: Bounding the cell by a rectangular box.

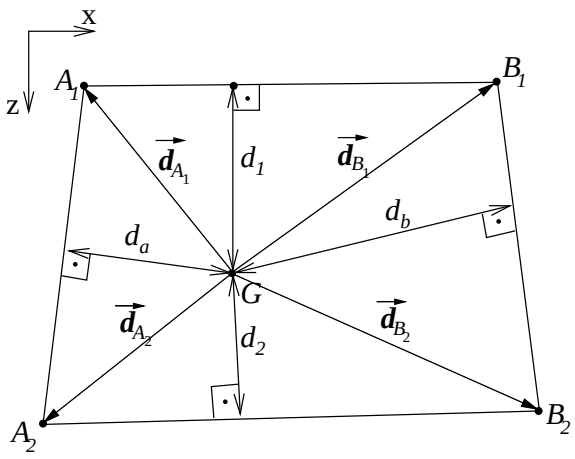

Figure 6.7: The vector products $\mathbf{d}_{A_1} \times \mathbf{d}_{B_1}$, $\mathbf{d}_{B_1} \times \mathbf{d}_{B_2}$, $\mathbf{d}_{B_2} \times \mathbf{d}_{A_2}$, and $\mathbf{d}_{A_2} \times \mathbf{d}_{A_1}$ are used for the decision if the gridpoint G is inside the cell $A_1B_1B_2A_2$. The distances d_1, d_2, d_a and d_b are used for the computation of the distance weights required for the traveltime estimation at G.

a lower ray density which implies large cells. Because linear interpolation in large cells is not sufficient, I propose a distance-weighted averaging of extrapolated traveltimes instead. The traveltimes are extrapolated from nodes to a gridpoint considering the wavefront curvature at the nodes (Figure 6.8). The extrapolated traveltime τ_{a_1} from the node A_1 to the gridpoint G is given by

$$\tau_{a_1} = \tau_{A_1} + \frac{\left|\overline{GG'}\right|}{\frac{1}{2}(v_G + v_{G'})}, \tag{6.1}$$

where τ_{A_1} is the traveltime at node A_1, G' is the projection of point G on the approximated wavefront with the radius R, $\left|\overline{GG'}\right|$ is the distance between G and G', v_G is the velocity at point G, and $v_{G'}$ is the velocity at point G'. The extrapolation of traveltimes from the other three nodes to the gridpoint G is similar, i.e., I obtain four traveltimes for gridpoint G.

To estimate the traveltime τ_G at gridpoint G, I weight the extrapolated traveltimes using the formula

$$\tau_G = w_{a_1}\tau_{a_1} + w_{a_2}\tau_{a_2} + w_{b_1}\tau_{b_1} + w_{b_2}\tau_{b_2}, \tag{6.2}$$

where w are distance weights. To ensure the continuity of estimated traveltimes, the distance weights are computed using the distances from the gridpoint to the ray segments and from the gridpoint to the wavefront segments, e.g., the weight w_{a_1} is proportional to $(1/d_1d_a)^k$, where d_1 is the distance to the wavefront segment $\overline{A_1B_1}$ and d_a is the distance to the ray segment $\overline{A_1A_2}$ (Figure 6.7), and k is a natural number. After testing different values of k, I decide to use $k = 2$. Multiplying all weights by $(d_1d_2d_ad_b)^k$, we obtain that

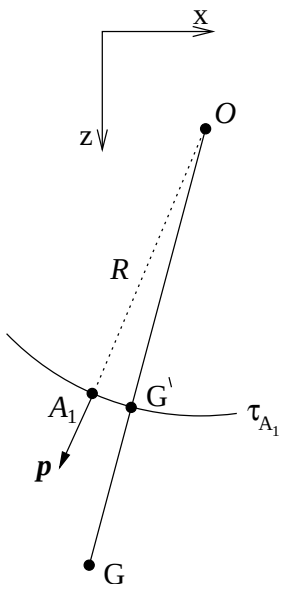

Figure 6.8: Extrapolation of the traveltime from the node A_1 to the gridpoint G. The point G' is the projection of gridpoint G onto the circular wavefront which passes the node A_1 (see text for details).

w_{a_1} is proportional to $(d_2 d_b)^k$. Taking into account that the sum of all distance weights should be 1, the distance weight w_{a_1} can be written as

$$w_{a_1} = \frac{(d_2 d_b)^k}{(d_1 d_a)^k + (d_2 d_a)^k + (d_1 d_b)^k + (d_2 d_b)^k}, \tag{6.3}$$

where d_2 is the distance of G to the wavefront segment $\overline{A_1 B_1}$ and d_b is the distance of G to the ray segment $\overline{B_1 B_2}$. The computation of these distances is efficient since it uses the same vector products which are needed to determine whether a gridpoint is located within the cell under consideration, e.g., $d_1 = |\mathbf{d}_{A_1} \times \mathbf{d}_{B_1}| / \left|\overline{A_1 B_1}\right|$. The calculation of the other distance weights is similar.

When one or both rays of the cell leave the model, the approximation of the wavefront curvature is not possible. In this special case, the traveltimes are estimated by considering only the normal to the wavefront (slowness vector, i.e., linear interpolation). This happens only at the borders of the model. The larger errors in gridpoint traveltimes caused by the linear interpolation are acceptable here.

6.3 Numerical examples

With numerical examples I want to show the influence of the most important WRT input parameters (time step of wavefronts, time-difference threshold, upper distance threshold UDT, lower distance threshold LDT) on the accuracy and the computational speed of the WRT technique. The results allow to propose some rules for the choice of suitable input parameters.

The accuracy of the proposed technique depends on the accuracy of the ray-tracing procedure, the accuracy of the insertion of new rays, and the accuracy of the estimation of traveltimes within cells.

Using the fourth-order Runge-Kutta method, the accuracy of the ray-tracing procedure can be increased by decreasing the time step of rays. The optimal time step for the integration should lead to a propagation of the corresponding ray in space. This time step is always smaller than the smallest spatial variation of the velocity of the model under consideration. For a fourth order Runge-Kutta method the traveltime errors introduced by the numerical integration of the kinematic ray tracing system can be considered to be negligible.

In the WRT technique, the errors connected with the insertion of a new ray are of the same order as the errors of the parent rays because no interpolation on initial condition is involved here.

The traveltime errors at gridpoints are introduced by the method used to estimate gridpoint traveltimes from the traveltimes at nodes. As described above, we use a method of second order to estimate the traveltimes at grid points. The accuracy of this estimation can be increased by increasing the ray and wavefront densities, i.e., by decreasing the cell size.

In the numerical examples, I quantify the accuracy of the estimated gridpoint traveltimes with respect to reference traveltimes by the mean traveltime error (expressed in milliseconds) and the percentage of arrivals with an absolute traveltime error larger than a predefined value (in this case 0.4 ms).

The total CPU-time mainly consists of the time required for the propagation of rays, the time for the approximation of the wavefront curvature at nodes, and the time for the estimation of traveltimes at gridpoints. The CPU-time needed for the propagation of rays depends on the number of rays, while the CPU-time needed for the approximation of the wavefront curvature depends on the number of cells. The CPU-time needed for the estimation of ray quantities depends on the density of the output grid. The traveltimes can be estimated on a fine grid, which can be directly used for Kirchhoff migration, or on a coarse grid. For practical applications, traveltimes are computed on a coarse grid and interpolation from the coarse grid to the fine grid is performed during the migration process. For this interpolation the hyperbolic interpolation (Vanelle and Gajewski, 2002) may be applied.

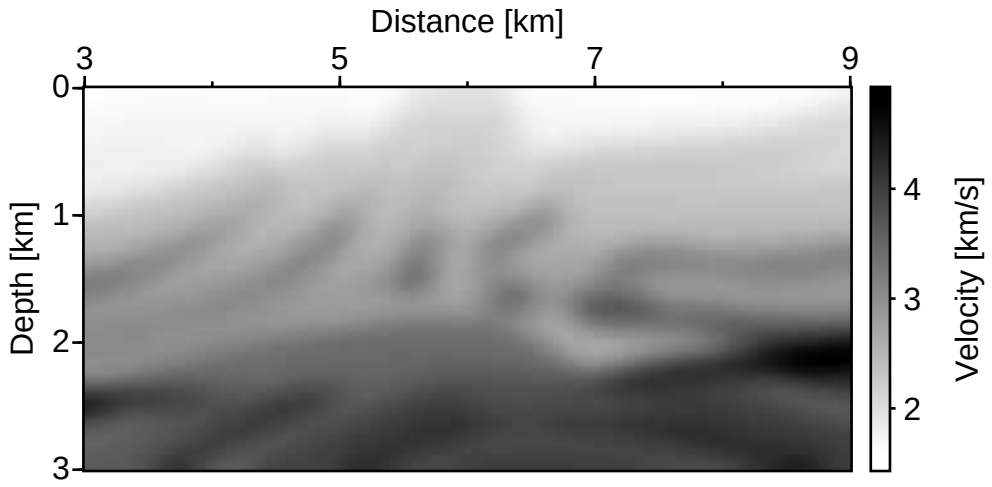

Figure 6.9: Smoothed version of the Marmousi model. Structural features of spatial dimension lower than 200 m are practically removed.

For the numerical examples, I use a fine output grid (grid distance 20 m). I chose a fine grid for a better graphical representation of the spatial distribution of the traveltime errors. In these examples, most of the CPU-time is used for the estimation of traveltimes on this fine grid. For practical applications, however, the traveltimes are estimated on a coarser grid (which can have, e.g., a factor of 100 less gridpoints), and in this case the number of rays and cells is the most important factor for the computational speed. Therefore, I quantify the computational speed by the number of rays and cells. Actually, the number of cells is a more sensitive criterion because it includes the number of rays and it also depends on the number of wavefronts.

Beside the accuracy and the computational speed, I also analyse the influence of the insertion criteria on the number of later arrivals which are found. This number depends on the ray density in caustic regions.

The first five computations are carried out in a smoothed version of the Marmousi model (Versteeg and Grau, 1991), where structural features of spatial dimension lower than 200 m are removed (Figure 6.9). The velocity grid has been resampled from 4 m to 20 m after smoothing. The source is located at coordinates $x = 6$ km and $z = 0$ km.

The first set of input parameters (Table 6.1) is designed to provide reference traveltimes (it is not possible to compute analytical traveltimes for complex models) and to estimate the number of later arrival traveltimes. Such parameters will not be used in practical applications. The time step of rays is set to 1 ms, the time step of wavefronts to 1 ms, the upper distance threshold UDT to 5 m, the lower distance threshold LDT to 0.1 m, and the

Data set number	1	2	3	4	5
Time step of rays [ms]	1	10	10	10	10
Time step of wavefronts [ms]	1	40	20	40	40
Upper distance threshold [m]	5	200	200	200	400
Lower distance threshold [m]	0.1	0	0	0	0
Time-difference threshold [ms]	0.1	4	4	8	4
Max. number of rays	16443	318	374	237	216
Number of cells	4567635	2466	5052	2206	1692
Number of gridpoints with later arrivals	15428	14850	14820	14143	13517
Mean traveltime error [ms]	-	0.09	0.06	0.10	0.39
Arrivals with errors > 0.4 ms [%]	-	2.8	1.3	2.9	26.2

Table 6.1: Five sets of input parameters and the resulting quantities used for the estimation of the efficiency of the WRT technique.

time-difference threshold to 0.1 ms. The resulting number of rays and cells is huge (see Table 6.1). Since even higher ray and wavefront densities did not noticeably change the values of the traveltimes, we define the traveltimes computed with this set of parameter as reference traveltimes (Fig. 6.10 shows corresponding isochrones).

The second set of input parameters (see Table 6.1) is more close to practical applications (except the LDT). In this example the time step of rays is 10 ms, the time step of wavefronts is 40 ms, UDT is 200 m, LDT is 0 m, i.e., LDT is not used, and the time-difference threshold is 4 ms. The resulting maximum number of rays is 318 and the number of cells is 2466. A number of 14850 later arrivals were found (96 % of the number of later arrivals which are computed with the first set of parameters). The decomposition of the model into cells is displayed in Fig. 6.11a. Note the uniform distribution of first-arrival rays and the high ray density in the caustic regions since the LDT was set to be 0. Most of the rays are inserted because the node-distance between adjacent rays becomes larger than 200 m. The spatial distribution of traveltime errors (Fig. 6.11b) follows both rays and wavefronts. The mean traveltime error is 0.09 ms. Except for the boundary regions where linear interpolation is applied, there is only a small region at an offset of 7 km and a depth of 2.2 km with an error larger than 2 ms. Fig. 6.11c shows that more than 70 % of the errors are smaller than 0.1 ms and only 2.8 % are larger than 0.4 ms.

For data set 3, I changed the time step of wavefronts to 20 ms and kept the other input parameters as for data set 2. As expected the number of cells gets larger, because the number of wavefronts is doubled. The decomposition of the model into cells is shown in Figure 6.12a. Because the distance between adjacent rays is larger than the distance between adjacent wavefronts, the spatial distribution of the errors follows the rays (Figure 6.12b). Compared to data set 2, the accuracy of the traveltime increases (the mean error decreases from 0.09 ms to 0.06 ms) and the computational speed decreases (the number of rays and cells increases, see Table 6.1). The computational speed can be in-

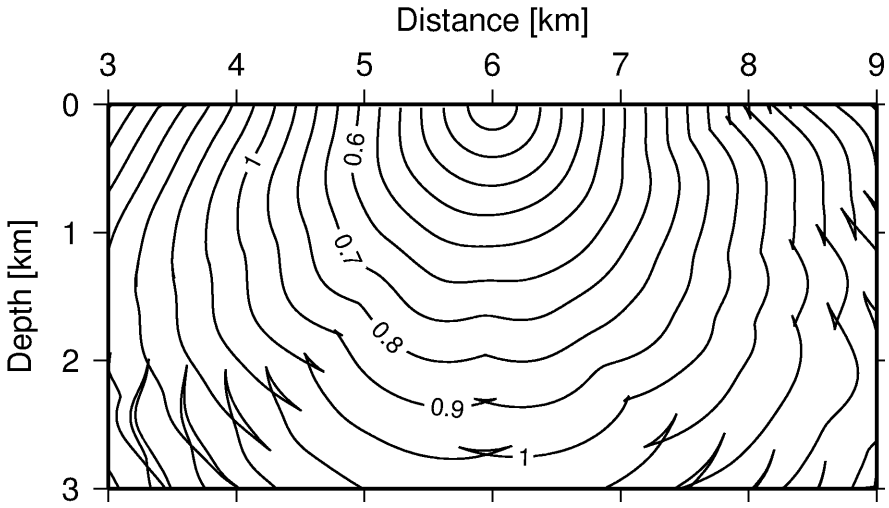

Figure 6.10: Isochrones plotted for data set 1, the reference traveltimes (see input parameters in Table 6.1)

creased by using a larger time step of wavefronts (e.g., 80 ms), but the accuracy (which is comparable to the accuracy obtained for data set 5) might not be sufficient for practical applications.

For data set 4, I changed the time-difference threshold to 8 ms and kept the other input parameters as for data set 2. Compared to data set 2, the number of rays decreases by 22 %, and the number of cells by 10 %. At a first glance there is no difference between the distribution of cells for data set 2 and for data set 4 (Figure 6.11a and Figure 6.13a). Moreover, the spatial and percentual distribution of the traveltime errors are nearly the same (Figure 6.11b and Figure 6.13b, respective Figure 6.11c and Figure 6.13c). However, a more careful inspection of the cell distribution shows differences in the caustic regions. As a consequence the number of later arrivals decreases by 5 % compared to data set 2.

For data set 5, I changed the UDT to 400 m and kept the other input parameter as for data set 2. Compared to data set 2, the number of rays decreases by 32 %, and the number of cells by 31 %. In contrast to data set 4, this decrease of rays and cells is connected with a sizeable increase of the traveltime error. The mean traveltime error is four times larger than in the second computation, and 26.2 % of arrivals have a traveltime error larger than 0.4 ms (Figure 6.14c). The spatial distribution of the error follows the rays. This example emphasises the importance of the UDT for the accuracy of the traveltimes.

The next two computations show the influence of the LDT on the computational speed of the WRT technique. These computations are carried out for a smoothed version of the Marmousi model, where variations of spatial dimension lower than 120 m are removed

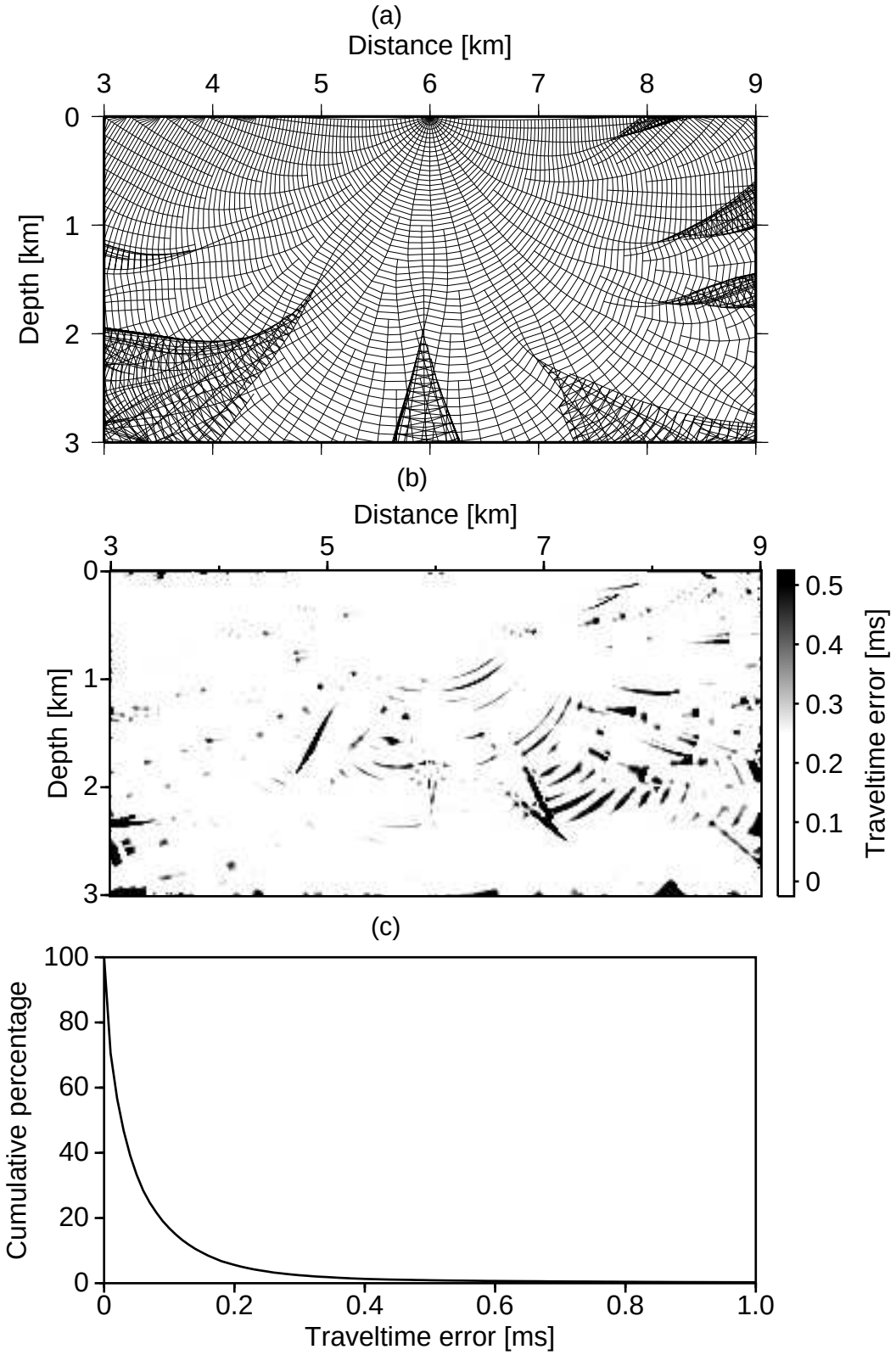

Figure 6.11: Results for data set 2 (see input parameters in Table 6.1). (a) Distribution of cells. Note that the rays are inserted by tracing from the source, but in this picture only the ray segments which bound cells are drawn. (b) Spatial distribution of the traveltime errors. (c) Distribution of the traveltime errors in percent (2.8 % of the traveltimes have an error larger than 0.4 ms).

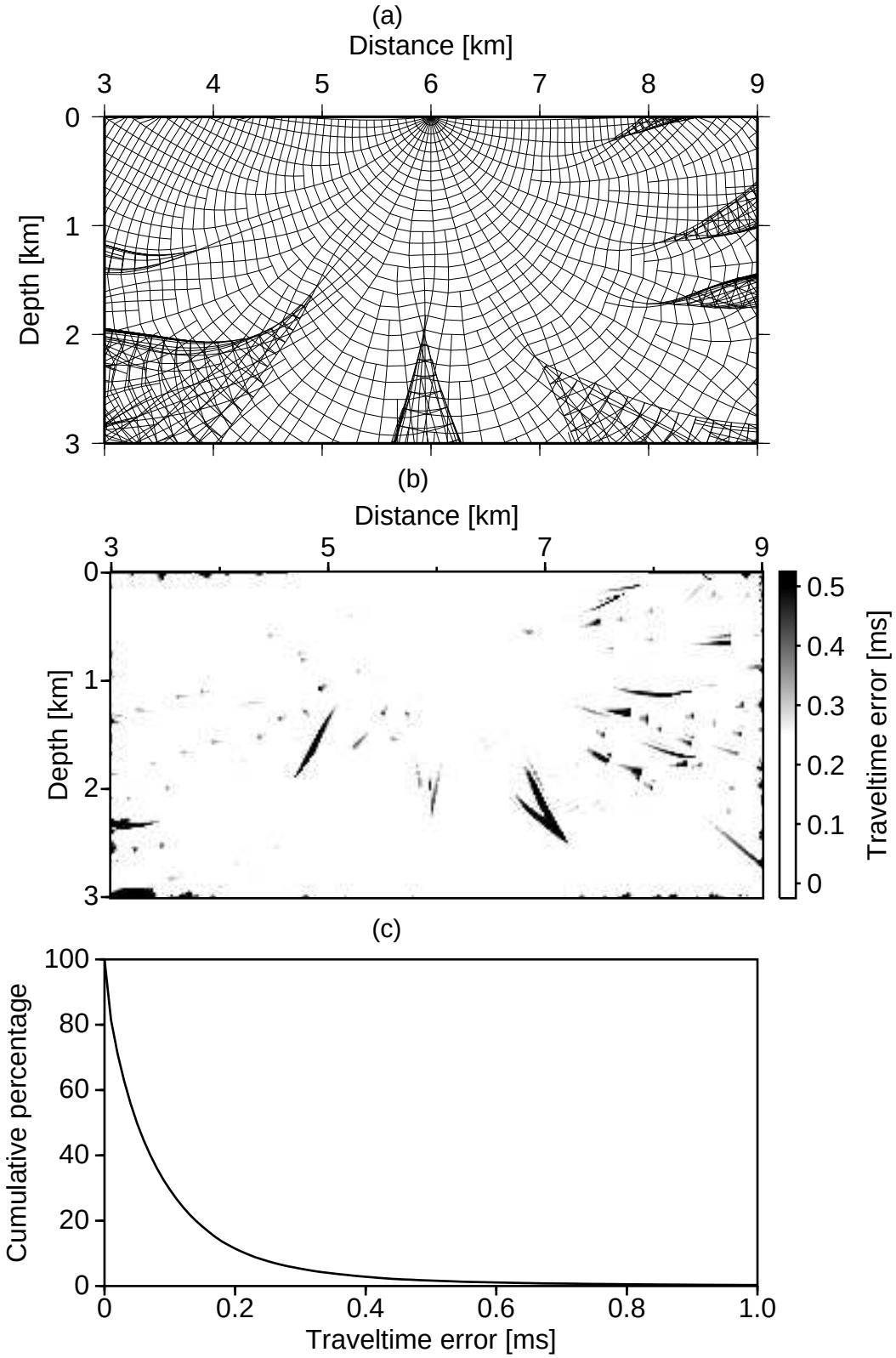

Figure 6.12: Results for data set 3 (see input parameters in Table 6.1). (a) Distribution of cells. (b) Spatial distribution of the traveltime errors. (c) Distribution of the traveltime errors in percent (1.3 % of the traveltimes have an error larger than 0.4 ms).

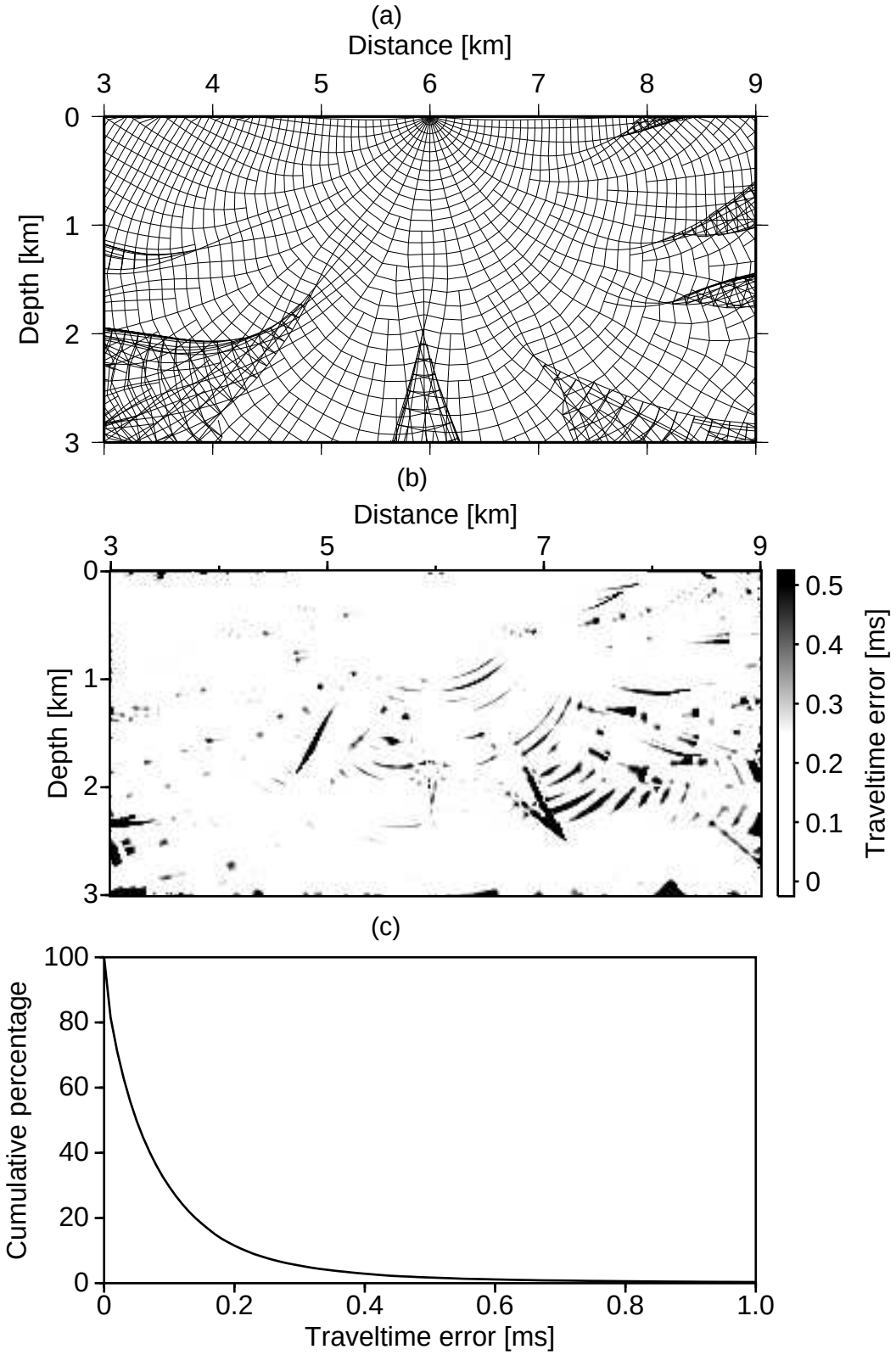

Figure 6.13: Results for data set 4 (see input parameters in Table 6.1). (a) Distribution of cells. (b) Spatial distribution of the traveltime errors. (c) Distribution of the traveltime errors in percent (2.9 % of the traveltimes have an error larger than 0.4 ms).

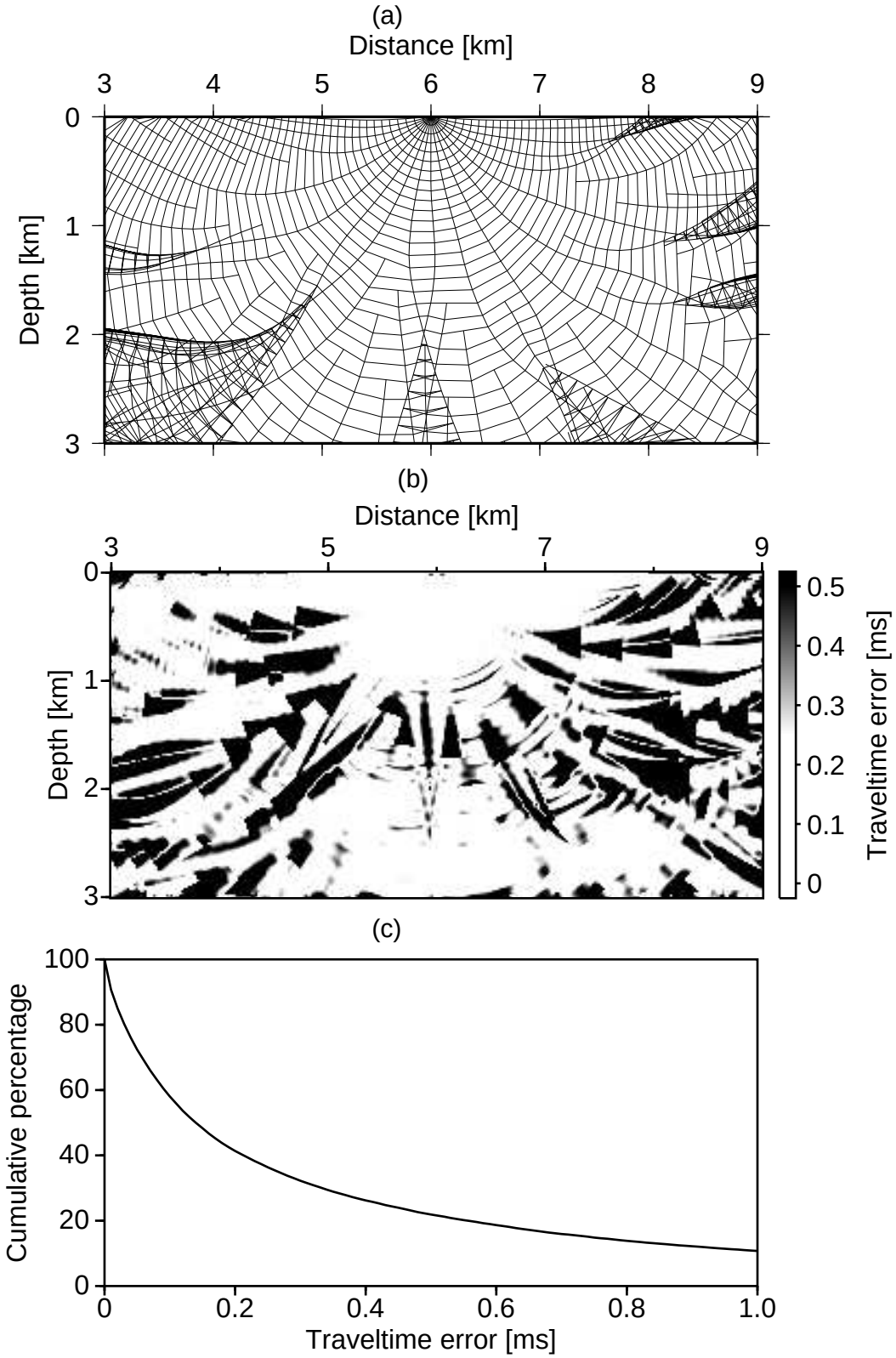

Figure 6.14: Results for data set 5 (see input parameters in Table 6.1). (a) Distribution of cells. (b) Spatial distribution of the traveltime errors. (c) Distribution of the traveltime errors in percent (26.2 % of the traveltimes have an error larger than 0.4 ms).

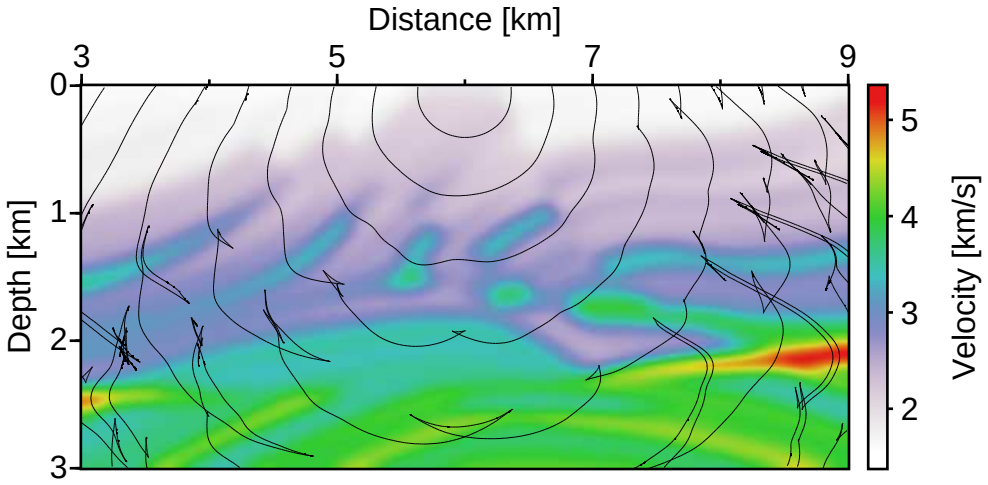

Figure 6.15: Smoothed version of the Marmousi model and isochrones. Structures of spatial dimension lower than 120 m are removed. The black lines are isochrones plotted with a contour interval of 0.2 s.

(Figure 6.15). The velocity grid was resampled from 4 to 12 m after smoothing.

The input parameters for sets 6 and 7 are given in Table 6.2. The only difference between the two sets of input parameters is the LDT, which is 0 m (i.e., it is not applied) for data set 6 and 1 m for data set 7. The influence of the LDT on the number of rays is considerable; the number of rays decreases from 3639 in data set 6 to 936 in data set 7. The distribution of rays resulting for data set 6 is shown in Figure 6.16, while the distribution of rays for data set 7 is shown in Figure 6.17. Because the decrease of the ray density affects only caustic regions, the accuracy of traveltimes in regular zones is not affected.

The numerical computations allows the following conclusions:

- The WRT technique allows an accurate computation of multivalued traveltimes with a relative small number of rays and wavefronts (compare data set 1 and 2). The accuracy of data set 2 is sufficient for practical applications (more than 99 % of the estimated traveltimes have errors which are smaller than 1 ms). A further increase of the accuracy does not justify the decrease of computational speed.

- The accuracy is mainly controlled by the time step of wavefronts and by the upper distance threshold. The accuracy decreases with the increase of the time step of wavefronts, the upper distance threshold, and the time-difference threshold.

- The computational speed is mainly controlled by the time step of wavefronts, by the upper distance threshold, and by the lower distance threshold. In principal, the

Data set number	6	7
Time step of rays [ms]	5	5
Time step of wavefronts [ms]	20	20
Upper distance threshold [m]	120	120
Lower distance threshold [m]	0	1
Time-difference threshold [ms]	4	4
Max. number of rays	3639	936
Max. number of cells	39695	16596

Table 6.2: The input parameters and the resulting number of rays and cells for the sixth and seventh computation.

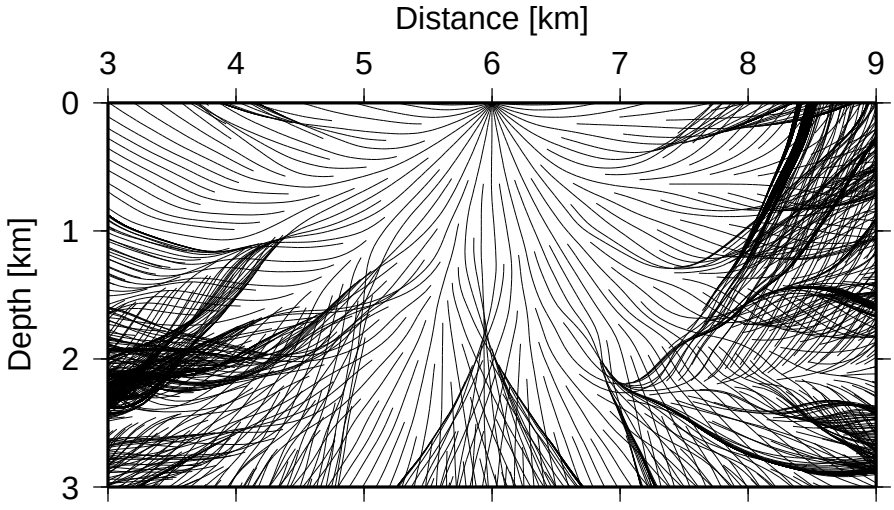

Figure 6.16: Distribution of rays obtained by setting the lower distance threshold to 0 m (data set 6). The number of rays is 3639. Note the oversampling by rays in caustic regions.

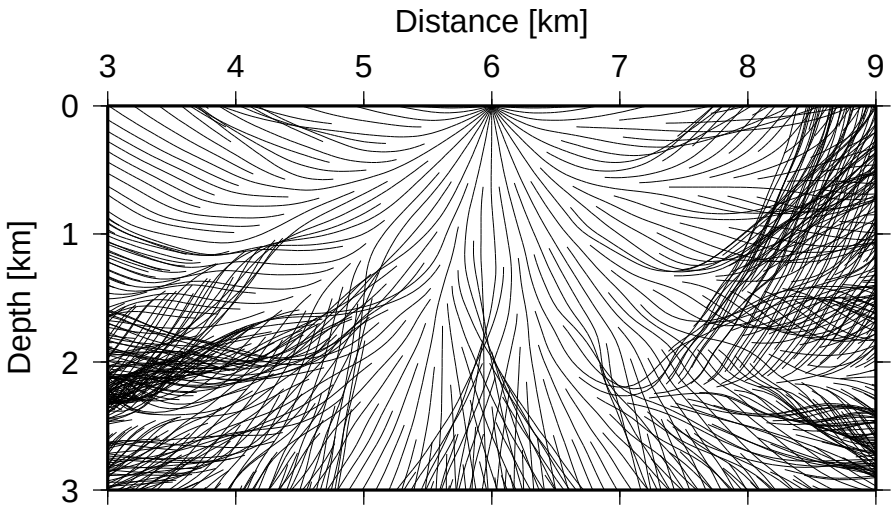

Figure 6.17: Distribution of rays obtained by setting the lower distance threshold to 1 m (data set 7). The number of rays is 936. The lower distance threshold changes the ray density only in caustic regions and avoids oversampling here.

increase of the computational speed is combined with the decrease of the accuracy. However, a reasonably small lower distance threshold (e.g., 1 m) increases the computational speed and has little influence on the accuracy of traveltimes.

- The spatial distribution of the absolute traveltime error follows both wavefronts and rays (uniform distribution), if the upper distance threshold is comparable with the spatial distance between two adjacent wavefronts. The choice of these distances should be related to the shortest spatial wavelength of the smoothed velocity model. A one-to-one relation might be a good guess for most applications. Smaller distances lead to oversampling by rays and wavefronts and are not justified by the increase in accuracy, while larger distances lead to poor illumination of small-scale velocity anomalies and to large errors.

6.4 Conclusions

For multivalued traveltime computation, I have presented the wavefront-oriented ray-tracing technique. This technique starts with few rays which are propagated stepwise through a smooth velocity model. To obtain a sufficient illumination throughout the model, I insert a new ray if certain criteria are satisfied. The new ray is inserted by tracing it directly from the source without any interpolation of initial conditions. The resulting ray field displays the same accuracy for the whole model. The insertion procedure avoids

the main source of errors in the WFC methods (errors connected with the insertion of new rays by interpolation on the wavefront). In the WRT technique, I further reduce other types of errors. The insertion criteria reduce the errors connected with small-scale velocity variations and with caustic zones. The distance-weighted averaging of extrapolated traveltimes reduces the errors of the traveltime even for large ray cells since the curvature of the wavefront is acknowledged.

The insertion of new rays by tracing them from the source and the estimation of traveltimes by considering the wavefront curvature allow the use of large distances between rays, i.e., a gain in computational speed is achieved because less rays and cells are used for the computation.

In Chapter 7, the method will be extended to three dimensions, where computational efficiency is even more important. The 2D investigation has provided the strategy for the efficient implementation of the 3D wavefront-oriented ray-tracing technique.

Chapter 7

3D wavefront-oriented ray tracing

7.1 Introduction

In Chapter 6, I have presented the 2D WRT technique as an efficient method for the computation of multivalued traveltimes on a coarse grid. For the 3D case the efficiency of the method is even more important because the number of traveltimes at gridpoints which need to be computed is significantly higher.

In this chapter, I present the 3D version of the WRT technique. In the following, I assume that the reader is familiar with the 2D WRT technique (Chapter 6) and the 3D WFC method (Chapter 4). All these algorithms are based on the same three steps:

1. The propagation of the wavefront
2. The insertion of new rays
3. The estimation of traveltimes within ray cells

The propagation of the wavefront in the 3D WRT and 3D WFC is similar, but because of the higher accuracy of the new inserted rays the 3D WRT starts with fewer rays (only two iterations at the source, i.e., 162 rays instead of 2562 rays). Therefore, in the following section I describe only the last two steps.

7.2 Description of the method

7.2.1 Approximation of the wavefront curvature

The wavefront curvature is used both for the insertion of new rays and for the estimation of traveltimes within ray cells.

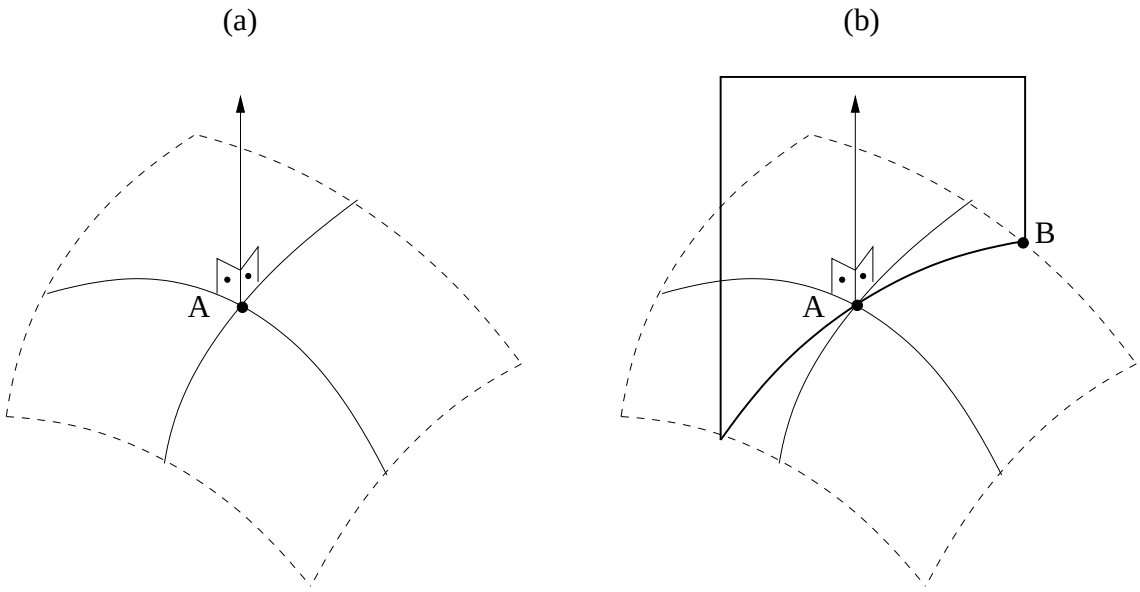

Figure 7.1: The surface of a wavefront in the vicinity of a given point A. (a) The solid curves are along the principal curvatures of this surface at point A. (b) The bold curve is a normal section. A normal section is the intersection of the surface with a normal plane. This plane is defined by the normal to the surface at point A and point B.

In 2D the wavefront is represented by a plain curve. The approximation of the radius and the centre of the wavefront curvature has been presented in Section 6.2.2. In 3D, the wavefront is represented by a surface (Figure 7.1a). The curvature of a surface at a given point A changes with the direction. This curvature is described by a curvature matrix. The eigenvalues of this matrix, K_1 and K_2, represent the principal curvatures of the surface at the given point. The principal radii of curvatures are expressed as $R_1 = 1/K_1$ and $R_2 = 1/K_2$. The curvature matrix for a wavefront can be computed by DRT (e.g., Červený, 2001).

The WRT technique (DRT is not applied) does not compute the principal radii of curvature, but it approximates the radii of curvature for several normal sections by KRT. Using the icosahedron as the basic network at the point source, a node is connected to five adjacent nodes (Figure 4.6), i.e., the radii of curvature are computed in five normal sections (directions) at each node (Figure 7.2a). However, only two of these five curvatures are simultaneously known. This is because the computation of the radii of curvature is done within ray cells (Figure 7.2b). The computed radii of wavefront curvatures are not used to estimate the principal radii of curvatures, but to estimate the radius of curvatures in the direction of gridpoints. The radius of curvature in the direction of a gridpoint is obtained by linear interpolation between adjacent radii of wavefront curvature (Figure 7.2c).

The normal section at node A in the direction of node B (Figure 7.1b) is the intersection of

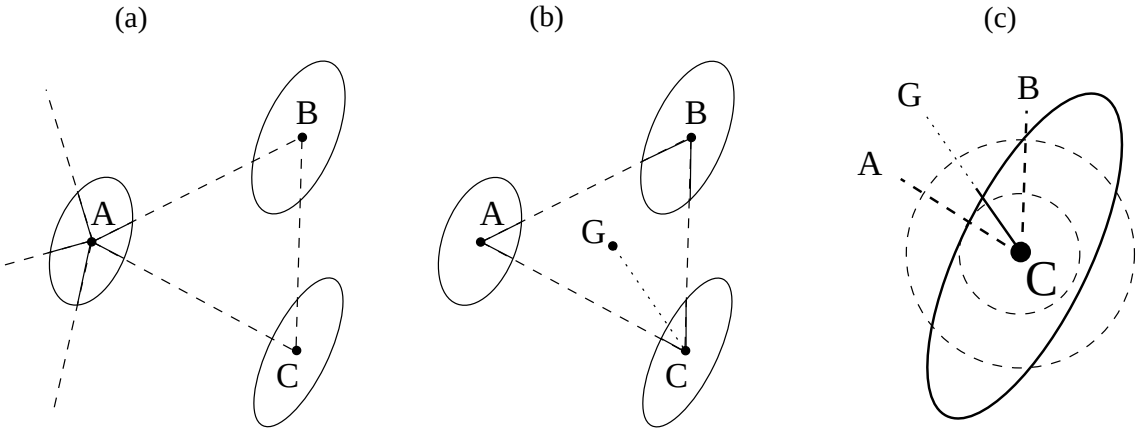

Figure 7.2: Wavefront curvatures in the 3D WRT technique. A, B and C are three nodes at the same wavefront. G is a gridpoint. The ellipses show the spatial distribution of the radii of wavefront curvature. (a) The wavefront curvature is computed in five directions at a node (see node A). (b) The curvatures are computed whitin cells, i.e., only two of these five curvatures are known simultaneously (for node A, the two curvatures are along the normal sections AB and AC). (c) The radius of wavefront curvature for an arbitrary direction (e.g., direction CG) is estimated by linear interpolation between the radii of wavefront curvatures in direction CA and CB.

a normal plane with the surface. This normal plane is defined by the normal to the surface at point *A* and point *B*. The estimation of the curvature of a normal section reduces to the 2D case and was described in Section 6.2.2.

Usually, the wavefront curvature at node *A* in the direction of node *B* differs from the wavefront curvature at node *B* in the direction of node *A*. Moreover, because the normal to the surface at point *A* is (usually) not coplanar with the normal to the surface at point *B*, the normal section at point *A* in the direction of point *B* will differ from the normal section at point *B* in the direction of point *A*. The difference between the radii of curvature which are determined in these two normal sections is used as a criterion for the insertion of new rays (insertion criterion 2, see next section).

7.2.2 Insertion of new rays

In this section, I present the approach used for the insertion of a new ray and the criteria used for the decision when to insert a new ray. Because a new ray is inserted between two adjacent rays, the insertion of a new ray in 3D is similar to the one in 2D (Section 6.2.3).

The initial direction of the new ray is the direction of the vector obtained by addition of the initial slowness vectors of the parent rays. For example, if the initial direction of

the first ray has a declination of 0° and an inclination of 90°, and the second ray has a declination of 90° and an inclination of 45°, then the new ray will have a declination of 35.26° and an inclination of 60°. The inclination and declination of the new ray are not the averages of the inclination and declination of the parent rays.

A new ray is inserted if one of the following insertion criteria is satisfied:

1. The distance between the adjacent nodes, e.g., distance $\overline{AB}$ in Figure 7.2, exceeds a predefined threshold (upper distance threshold, UDT).
2. The difference in wavefront curvature at adjacent nodes exceeds a predefined threshold (time-difference threshold, see Section 6.2.3) and the distance between the nodes is larger than a predefined lower distance threshold (LDT).
3. One ray in the cell crosses the surface defined by the other two adjacent rays and the distance between the nodes is larger than the LDT.

The first two criteria are the same as the first two ones in the 2D case. The third insertion criterion is similar to the third one in 2D, but the implementation is different. In 2D the two adjacent rays cross each other in the caustic region (Figure 7.3a), while in 3D one ray crosses the area defined by the other two adjacent rays (Figure 7.3b). In both cases (2D and 3D) it is ambiguous which ray has passed the caustic. Because of this ambiguity, the third insertion criterion (3D case) inserts a new ray between each of the three pairs of adjacent rays in that cell.

The reasons for using these insertion criteria were already given in Section 6.2.3. To insert a new ray, I trace it directly from the source. The advantages of this approach are also described in Section 6.2.3.

An optimal illumination by rays is a precondition for an accurate estimation of traveltimes within cells. An optimal illumination allows the computation of sufficiently accurate traveltimes from a minimum number of rays. The estimation is described in the next section.

7.2.3 Estimation of traveltimes within cells

For the estimation of traveltimes within 3D ray cells, three algorithms have been sufficiently described in the literature. These algorithms are proposed by Vinje et al. (1996a), by Lucio et al. (1996), and by Bulant and Klimeš (1999). Because the estimation approach is closely connected with the assignation of a gridpoint to a cell (assignation approach), I will present these approaches all together.

Vinje et al. (1996a) project the gridpoint to the old wavefront (Figure 4.1). If the projection point lies within (or very close to) the base triangle (Figure 4.3), they assign the

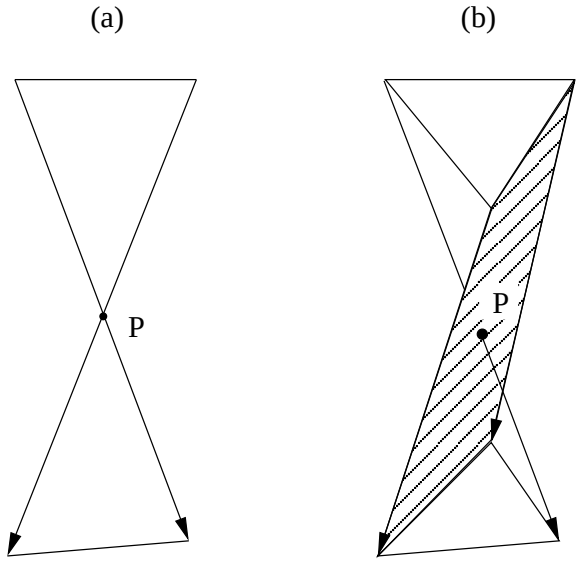

Figure 7.3: (a) Two rays cross each other (2D case). (b) A ray crosses the surface defined by other two rays (3D case). Point P denotes the crossing point.

gridpoint to the cell. This approach allows the assignation of a gridpoint to more then one cell, i.e., the same arrival is estimated several times. To avoid multiple storage of the same arrival, Vinje et al. (1996a) use the take-off direction: if the take-off direction of two estimated arrivals is nearly the same, then only one arrival is stored. Because of errors in the estimation of the take-off direction, this approach might fail. Moreover, the multiple estimation of the same arrival decreases the computational speed.

Lucio et al. (1996) divide the cell into three tetrahedra. Because the assignation of the gridpoints to a tetrahedron is unambiguous, they avoid multiple estimation of the same arrival. For the estimation of traveltimes within a tetrahedron, they use a linear interpolation.

Bulant and Klimeš (1999) proposed a more accurate bicubic interpolation for traveltimes, but their approach for the assignation of a gridpoint to a ray cell is not as efficient as the one proposed by Lucio et al. (1996). For the assignation of a gridpoint to a cell, Bulant and Klimeš (1999) construct a plane which contains the given gridpoint. The intersection of this plane with the ray segments builds a triangle. The decision whether the gridpoint is within the cell is replaced by the decision whether the gridpoint is within the triangle. The construction of the plane is time-consuming because a cubic equation has to be solved. Moreover, the approach does not define unambiguous lateral sides. The ambiguity leads to difficulties for the assignation of gridpoints close to the cell's boundary.

For the unambiguous assignation of a gridpoint to a cell, I use the approach proposed by Lucio et al. (1996) (see also Section 4.1.3). For the estimation of the traveltimes

within a cell, I propose a distance-weighted averaging of extrapolated traveltimes. This second-order estimation approach (which is more accurate than linear interpolation) will be described below.

Flowchart of the algorithm

The flowchart of the algorithm is summarised in Figure 7.4. In the numerical implementation, the subroutine for the estimation of traveltimes within cells follows the subroutine for the propagation of wavefronts. Each cell between two wavefronts has its unique number. Every cell is tested whether it is a caustic cell. If this is the case then the next cell is considered; if not, the cell is bounded by a rectangular box. To detect caustics, I use the same approaches as in the WFC method (Section 4.1.3).

Because the 3D WRT technique does not compute the KMAH index for each ray, it is not possible to detect caustic cells if a ray does not cross the opposite lateral cell side. On the other hand, the third insertion criterion increases the ray density in caustic regions. A high ray density corresponds to very small caustic cells, i.e., few gridpoints are placed in these cells. Moreover, the estimation of traveltimes within very small cells is accurate even in caustic cells. The estimation of amplitude related ray quantities (e.g., relative geometrical spreading) is not accurate; however the poor accuracy of amplitudes in caustic regions is a general problem of ray-tracing techniques, not only of the WRT technique.

For cells which have all six nodes within the model, the wavefront curvature is approximated for each node along two normal sections. A node belongs to five adjacent rays (see Figure 4.4), i.e., the wavefront curvature is approximated along five normal sections. However, because this approximation is done within cells, only two of this five curvatures are known simultaneously (see Section 7.2.1). The next step is the loop over the gridpoints of the bounding box. If the gridpoint is within the cell, then the traveltimes are extrapolated from nodes to the gridpoint. In a six-node cell, the traveltimes are extrapolated using the wavefront curvature, otherwise the traveltimes are extrapolated using the slowness vector, i.e., linear interpolation. The extrapolated traveltimes are then weighted considering the distances to cell sides and to the ray segments.

In case of multivalued arrivals, the storing is preceded by sorting. The arrivals are sorted with regard to their traveltimes. If the gridpoint was not the last in the bounding box, then a new gridpoint is considered, otherwise a new cell is considered and the procedure is repeated.

Extrapolation of traveltimes

For cells which have all six nodes within the model, the radius and the centre of the wavefront curvature are approximated for each node along two normal sections (see Figure 7.2b). The approach which is used for this approximation is the same as in the 2D case

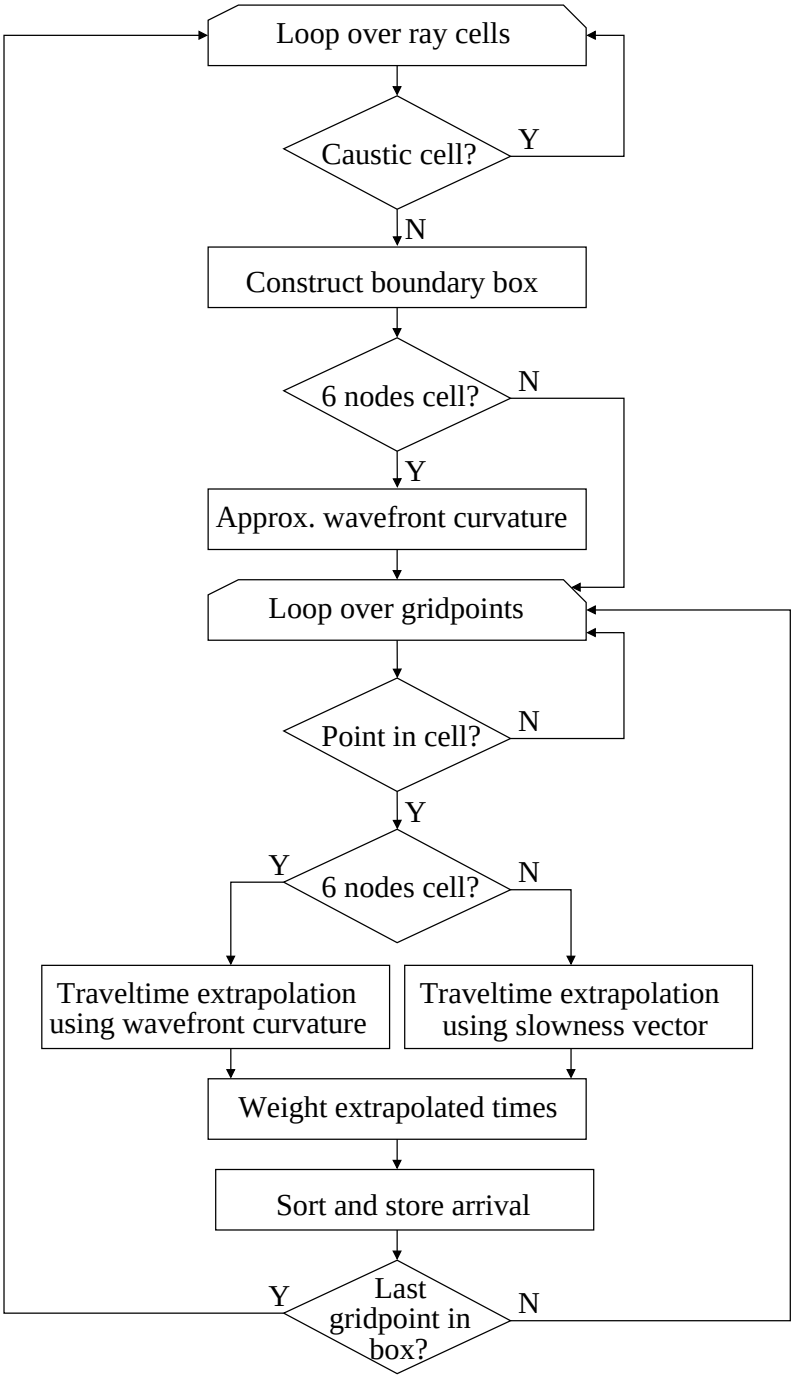

Figure 7.4: Flowchart for the estimation of traveltimes within ray cells.

(see Section 6.2.2). To each normal section I assign a sphere defined by the above mentioned centre and radius of the wavefront curvature. To extrapolate the traveltime from a node to a gridpoint, I use the distances between the gridpoint and these spheres. (In 2D, there are distances between the gridpoint and circles, see Section 6.2.4.) Each distance is transformed to a time by dividing this distance by a velocity value. This velocity is the average of the velocities at the node and the gridpoint. The extrapolated traveltime at the gridpoint is the traveltime at the node corrected by the above mentioned time value.

To each gridpoint I extrapolate twelve traveltimes. Because the confidence in the extrapolated traveltimes depends on different distances, I estimate the traveltime at a gridpoint by using distance-weights.

Distance weights in 3D cells

The traveltime t_G at the gridpoint G is estimated by the weighted average of the twelve extrapolated traveltimes (see also Figure 7.5):

$$t_G = w_{a_1\beta} t_{a_1\beta} + w_{a_1\gamma} t_{a_1\gamma} + w_{b_1\alpha} t_{b_1\alpha} + \ldots, \tag{7.1}$$

where $t_{a_1\beta}$ is the extrapolated traveltime from node A_1 to gridpoint G using the wavefront curvature at node A_1 along the normal section A_1C_1. The weight function $w_{a_1\beta}$ reflects the confidence in the extrapolated traveltime.

The extrapolation is most accurate along the ray which passes the node. The extrapolation is also good in the normal plane where the wavefront curvature has been estimated and in the triangular base of the node. The confidence in the extrapolated traveltimes can be expressed as a function of three distances:

- The distance between the gridpoint and the linear ray segment, e.g., d_a
- The distance between the gridpoint and the lateral side, e.g., d_β
- The distance between the gridpoint and the triangular base, e.g., d_1

The weight function $w_{a_1\beta}$ for the extrapolated traveltime $t_{a_1\beta}$ follows as

$$w_{a_1\beta} \sim \left(\frac{1}{d_1} \frac{1}{d_a} \frac{1}{d_\beta} \right)^k, \tag{7.2}$$

where the meaning of the distances d_1, d_a, d_β is shown in Figure 7.5. The superscript k is an integer which tunes the influence of the weights. Setting $k = 0$ means the same weights for all extrapolated traveltimes. On an empirical basis, I choose $k = 2$.

The weight function not only reflects the confidence in the accuracy of extrapolated traveltimes, but also ensures the continuity of the estimation across the cell boundaries.

The relation (7.2) gives the proportionality relation of the distance weight $w_{a_1\beta}$ before normalisation. After scaling all the weights with the factor $\left(d_1 d_2 d_a d_b d_c d_\alpha d_\beta d_\gamma\right)^k$, the relation (7.2) has the form

$$w_{a_1\beta} \sim \left(d_2 d_b d_c d_\alpha d_\gamma\right)^k.$$

Using the notation

$$w_1^* = \left(d_2 d_b d_c d_\alpha d_\gamma\right)^k$$

and extending this notation in a similar way to all twelve weights, the normalised weight is given by

$$w_{a_1\beta} = \frac{w_1^*}{\sum_{i=1}^{12} w_i^*}.$$

The expressions for the other weights are similar.

The distance between the gridpoint G and the ray segment is computed using the cross-product between the vectors which start from G and end at the nodes which define the ray segment. The distances between the gridpoint G and the triangular bases, and between the gridpoint and the lateral sides have already been computed for the assignation of the gridpoint to the cell.

To split a cell into three tetrahedra, a lateral side is approximated by two plane triangles. The distance between a gridpoint and a lateral side is approximated by the distance between the gridpoint and one triangle or by the average distance between the gridpoint and the two triangles. Let us consider that the gridpoint is within the tetrahedron $A_1A_2B_2C_2$ (Figure 7.5). In this case the distance to the lateral side β will be replaced by the distance to the plane which is defined by the nodes A_1, A_2 and C_2; the distance to the lateral side γ will be replaced by the distance to the plane $A_1A_2B_2$; and the distance to the side α will be replaced by the average of the distances to the planes $B_1C_2C_1$ and $B_1C_2B_2$.

7.2.4 Estimation of other ray quantities

Not only traveltimes, but also other ray quantities (e.g., slowness, take-off angle) are estimated by a distance-weighted averaging. However, in contrast to the traveltimes, the other ray quantities are not extrapolated, i.e., a linear interpolation is used. The distance weights use the same distances, but in different combination. The number of needed distance weights also differs: twelve for the traveltime, six for the slowness, three for the take-off angles, and two for the relative geometrical spreading. The distance weights for the take-off angles and for the relative geometrical spreading can easily be computed from the distance weights for the slowness.

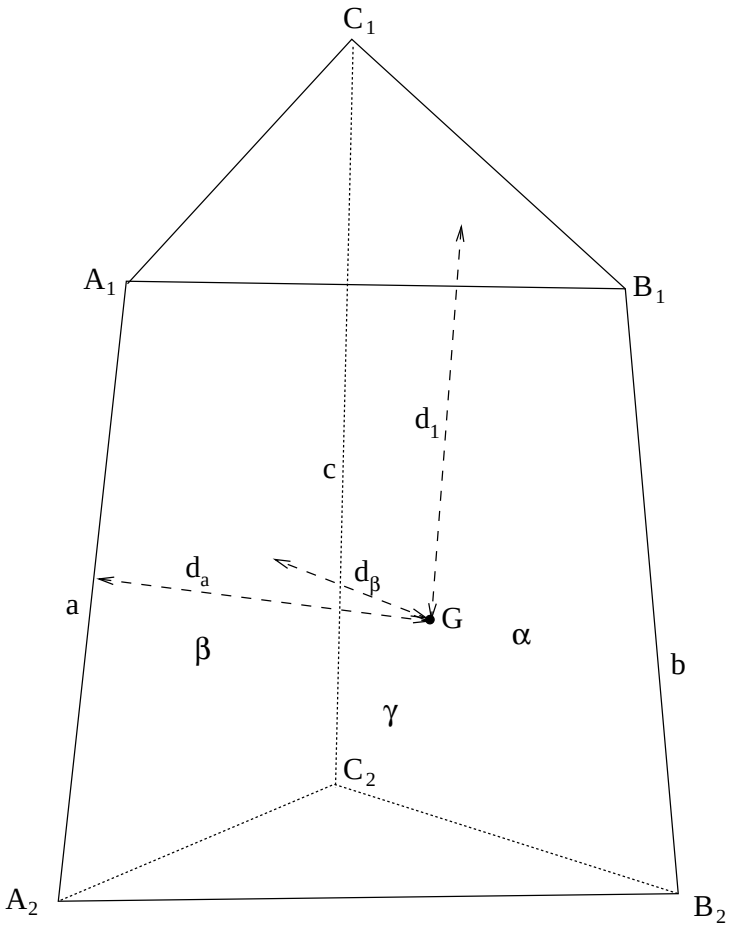

Figure 7.5: 3D cell and the distances which are used for the computation of the distance weight $w_{a_1\beta}$. d_1 is the distance between gridpoint G and the upper triangular base, d_a is the distance between G and the ray segment A_1A_2, and d_β is the distance between G and the side β.

Estimation of slowness components

The slowness components p_i at a gridpoint G are estimated by the formula

$$(p_i)_G = w_{A_1}(p_i)_{A_1} + w_{B_1}(p_i)_{B_1} + w_{C_1}(p_i)_{C_1} + w_{A_2}(p_i)_{A_2} + w_{B_2}(p_i)_{B_2} + w_{C_2}(p_i)_{C_2}. \quad (7.3)$$

The proportionality relation for the distance weight w_{A_1} is expressed as

$$w_{A_1} = \frac{1}{d_1}\frac{1}{d_a}\frac{1}{d_\beta d_\gamma}. \quad (7.4)$$

There are some differences between the relations (7.2) and (7.4). The distance weight $w_{a_1\beta}$ is used for the extrapolated traveltime from node A_1 to the gridpoint G considering the wavefront curvature in the normal section A_1C_1. The distance weight w_{A_1} is used for the slowness computed at node A_1. The node A_1 belongs to the lateral sides β and γ. There is no special connection to one of these sides, and therefore the distance weight (7.4) depends on both sides. Another difference between the two distance weights is the superscript k, which is 2 in relation (7.2) and 1 in relation (7.4). Setting the superscript k to 1 is motivated by the assumption of linear variations of the slowness components.

Following the same steps as presented for the distance weights used for the estimation of traveltimes, we can write the normalised distance weights used for the interpolation of slowness components as

$$w_{A_1} = \frac{w_1^*}{\sum_{i=1}^{6} w_i^*}. \quad (7.5)$$

In this case $w_1^* = d_2 d_b d_c d_\alpha$. The distance weights connected to other nodes are similar.

The incidence angles (local inclination and declination of the ray) are derived from the slowness components.

Estimation of the take-off angles

The take-off angles (inclination and declination of the ray at the source) are derived from the take-off slowness components. Because the take-off angles are constant along a ray, the number of distance weights reduces to three. The take-off slowness components p_{Si} at a gridpoint G are estimated by the formula

$$(p_{Si})_G = w_a(p_{Si})_a + w_b(p_{Si})_b + w_c(p_{Si})_c, \quad (7.6)$$

where

$$\begin{aligned} w_a &= w_{A_1} + w_{A_2}, \\ w_b &= w_{B_1} + w_{B_2}, \\ w_c &= w_{C_1} + w_{C_2}. \end{aligned} \quad (7.7)$$

Estimation of the relative geometrical spreading

The geometrical spreading along the ray can be expressed in terms of the principal radii of curvature (e.g., Červený et al., 1977). Since the principal radii of curvature are not known in the WRT technique (see Section 7.2.1), we cannot compute the geometrical spreading along the ray.

The relative geometrical spreading L (see Appendix C) along a ray tube can be approximately computed with the formula

$$L^2 \approx \frac{\left|\hat{\mathbf{x}}^{(21)} \times \hat{\mathbf{x}}^{(31)}\right|}{\left|\hat{\mathbf{p}}_S^{(21)} \times \hat{\mathbf{p}}_S^{(31)}\right|}. \tag{7.8}$$

where the vector $\hat{\mathbf{x}}^{(21)}$ is the difference between the position vectors $\hat{\mathbf{x}}^{(2)}$ and $\hat{\mathbf{x}}^{(1)}$. Similarly, $\hat{\mathbf{x}}^{(31)} = \hat{\mathbf{x}}^{(3)} - \hat{\mathbf{x}}^{(1)}$ (Figure 5.2). The quantity $\left|\hat{\mathbf{p}}_S^{(21)} \times \hat{\mathbf{p}}_S^{(31)}\right|$ is the same for the whole ray tube and needs to be computed only once. This quantity represents the solid angle at the source divided by the squared velocity at the source position. Please note that the relative geometrical spreading is a byproduct of the detection of caustics by monitoring the changes of the direction of the normal to the triangular bases (Section 5.2).

The relative geometrical spreading L at gridpoint G is estimated by linear interpolation between two triangular bases, i.e.,

$$L_G = w_1 L_1 + w_2 L_2, \tag{7.9}$$

where L_1 and L_2 are assigned to the old and new triangular bases, and the distance weights w_1 and w_2 are computed by the relations

$$\begin{aligned} w_1 &= w_{A_1} + w_{B_1} + w_{C_1}, \\ w_2 &= w_{A_2} + w_{B_2} + w_{C_2}. \end{aligned} \tag{7.10}$$

7.3 Numerical examples

To show the accuracy of the distance-weighted averaging of extrapolated traveltimes, I use two models which allow analytical computation of traveltimes. The first one is a homogeneous velocity model, the second one is a constant velocity gradient model. The geometry of the model is the same in both cases (Figure 7.6). The velocity in the homogeneous model is 2000 m/s. The velocity distribution in the constant velocity gradient model

is given by $v(z) = 2000 + 0.5z$, where $v(z)$ is the velocity at the depth z. Most results are shown for two vertical sections at $y = 2$ km and at $x = 2$ km respectively (Figure 7.6).

In the above mentioned homogeneous model I also test the accuracy of the inclination and declination angles and the accuracy of the relative geometrical spreading.

To show the performance of the 3D WRT technique, I have applied it to a 3D variant of the Marmousi model. Finally, I proof that the insertion of new rays by tracing them from the source not only increases the accuracy of the WRT technique but also decreases the computational time.

Homogeneous Model

In the first example, I compare the distance-weighted averaging of extrapolated traveltimes to the linear interpolation of traveltimes between plane triangular sides. These sides approximate the wavefronts. For the propagation of the wavefront in the WRT technique a time step of wavefronts of 0.1 s is used. A new ray is inserted if the distance between adjacent rays becomes larger than 500 m.

The wavefronts constructed from analytical traveltimes are shown in Figures 7.7a and 7.7b. The traveltime errors due to the linear interpolation are shown in Figures 7.7c and 7.7d, and the errors due to the distance-weighted averaging of extrapolated traveltimes are shown in Figures 7.7e and 7.7f. Note that even in the homogeneous velocity model the linear interpolation leads to errors up to 10 ms. The traveltime errors when using the distance-weighted averaging of extrapolated traveltimes are much smaller (less than 10^{-3} ms).

In the second example, I analyse the accuracy of the inclination and declination angles and of the relative geometrical spreading. For this example, I insert a new ray if the distance between rays becomes s larger than 200 m.

In the homogeneous velocity model, the direction of the ray does not change during the propagation of the ray. Therefore, the inclination and declination angles at the source are the same as the inclination and declination angles in the model. Figure 7.8a shows analytical results of the inclination angle in a vertical section at $x = 2$ km. The absolute errors of the computation of inclination angle by the WRT technique are shown in Figure 7.8b. The maximum error was 2°. The average error in this vertical section is less than 0.1°. The average error in the whole cube is also less than 0.1°.

Figure 7.9a shows analytical results of the declination angle in a horizontal section at $z =$ 1 km. The absolute errors of the computation of inclination angle by the WRT technique are shown in Figure 7.9b. The maximum error was 5°. The average error in this vertical section is less than 0.3°. The average error in the whole cube also is also less than 0.3°.

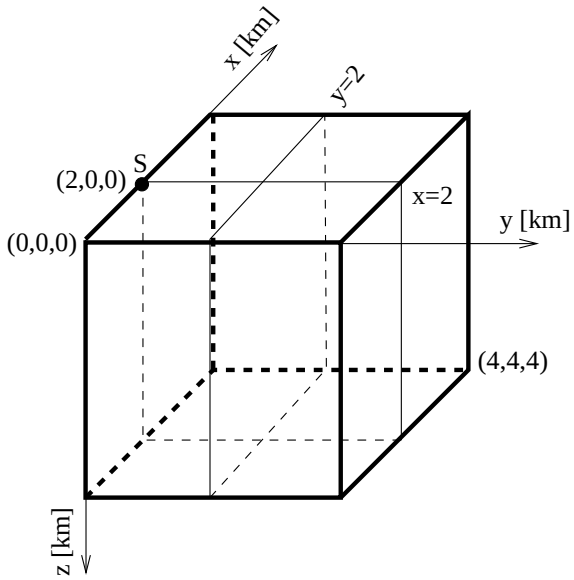

Figure 7.6: The cube model with the side length of 4 km. The source S is located at the coordinates (2 km, 0 km, 0 km). The results of the computations are shown for two vertical slices at $y = 2$ km and $x = 2$ km.

Figure 7.10a shows the relative geometrical spreading computed by the 3D WRT technique in a vertical section at $x = 2$ km. The exact computation of the relative geometrical spreading is shown in Figure 7.10b.

In the homogeneous velocity model, the relative geometrical spreading is proportional to the area of a spherical triangle divided by the corresponding solid angle at source. In the WRT technique, the area of the spherical triangle is approximated by the area of a plane triangle, and this approximation is poor near the source point. This poor approximation close to the source leads to large absolute and relative errors (Figure 7.10c,d). The region with large relative errors (larger than 3 %) of the relative geometrical spreading is located within a sphere with a radius of 0.5 km around the source point. The average relative error in the whole model is less than 0.1 %.

Figure 7.11a shows the relative geometrical spreading computed by the 3D WRT technique in a vertical section at $y = 2$ km. The exact computation of the relative geometrical spreading is shown in Figure 7.11b. Figure 7.11c and Figure 7.11d show the distribution of the absolute and relative error of the relative geometrical spreading.

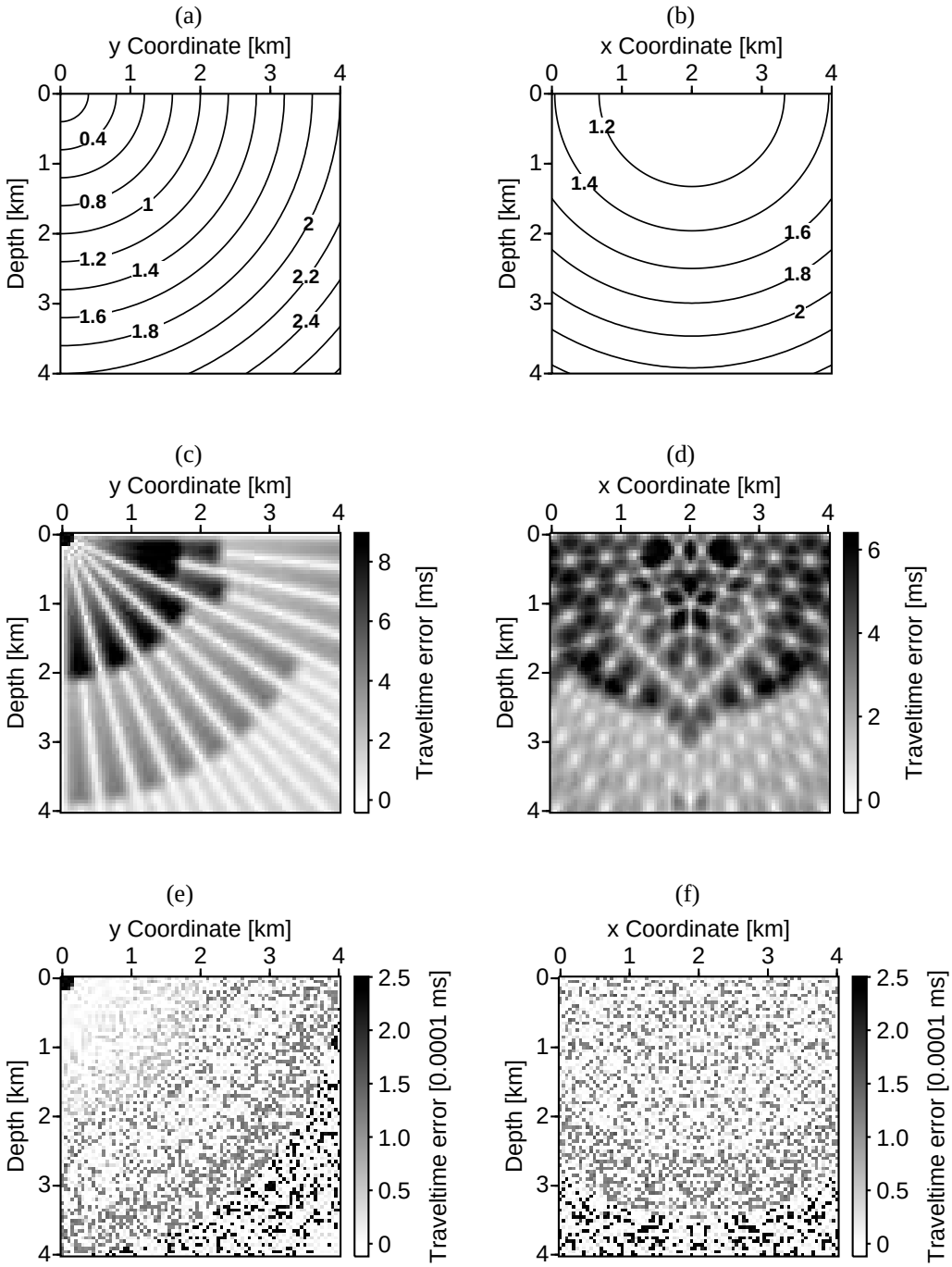

Figure 7.7: Results of the computation in the homogeneous velocity model in two vertical slices: (a), (c) and (e) for the vertical slice at $y = 2$ km; (b), (d) and (f) for the vertical slice at $x = 2$ km. (a) and (b): Wavefronts constructed from analytically computed traveltimes. (c) and (d): Traveltime errors due to the linear interpolation. (e) and (f): Traveltime errors due to the distance-weighted averaging of extrapolated traveltimes. Note the different error scales.

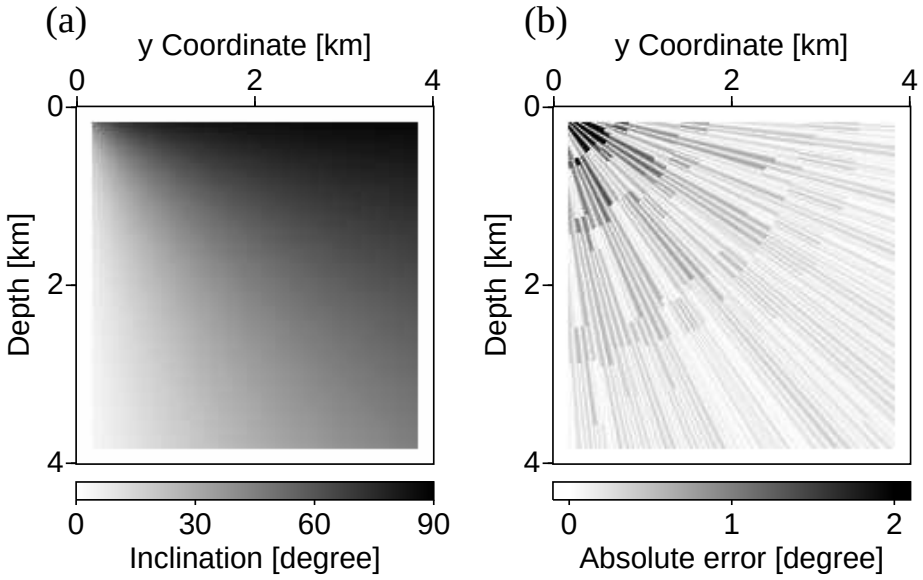

Figure 7.8: (a) Inclination angle and (b) absolute error of the inclination angle. The computation is done in a homogeneous velocity model. The geometry of the model is shown in Figure 7.6. The vertical section displayed here is at $x = 2$ km.

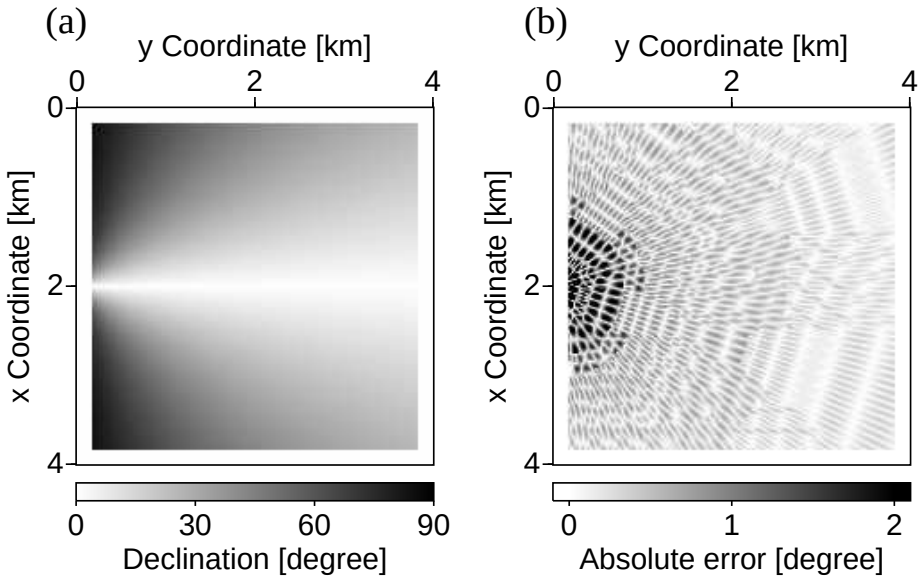

Figure 7.9: (a) Declination angle and (b) absolute error of the declination angle. The computation is done in a homogeneous velocity model. The geometry of the model is shown in Figure 7.6. The horizontal section displayed here is at $z = 1$ km.

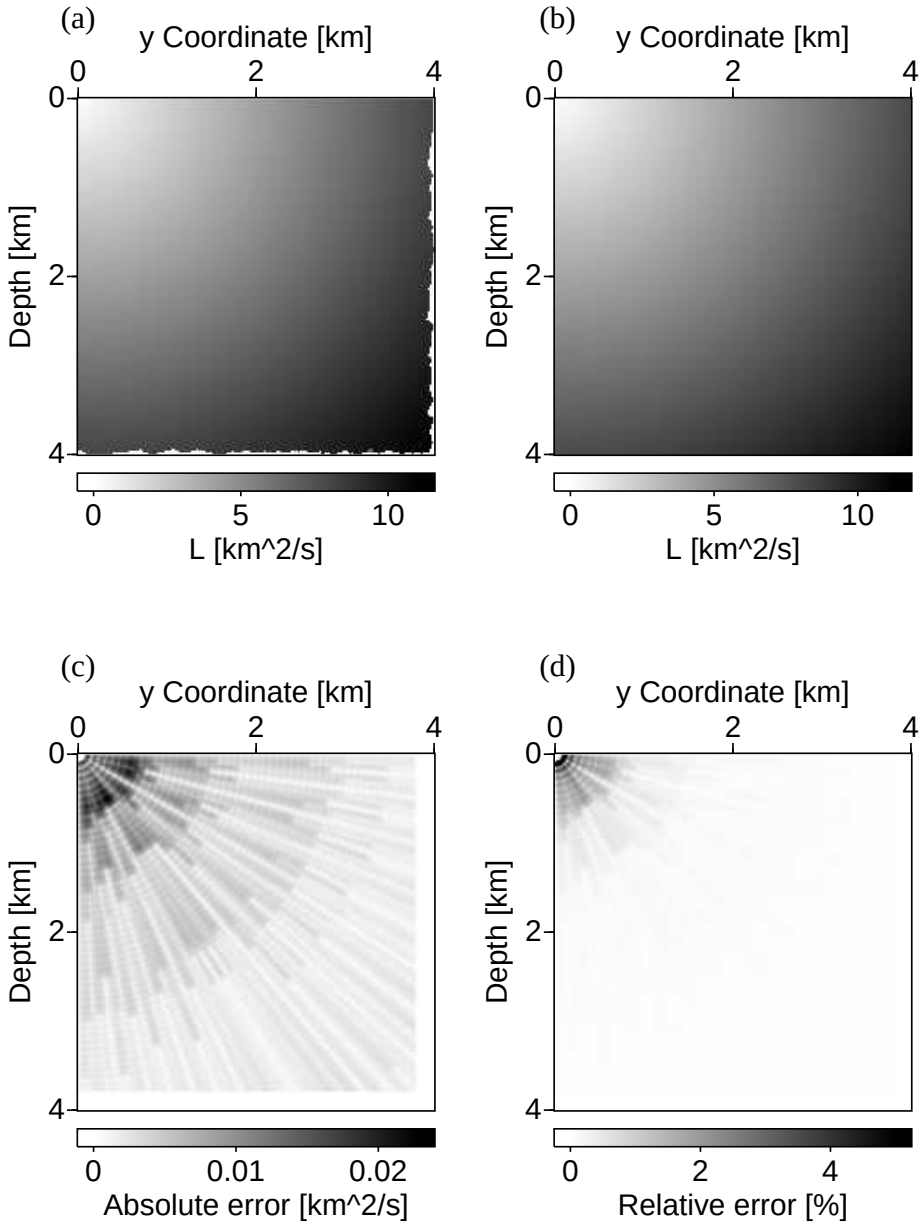

Figure 7.10: Relative geometrical spreading (L) in a homogeneous velocity model in the vertical section at $x = 2$ km. (a) computed by the WRT technique (b) analytical computation (c) absolute error (d) relative error.

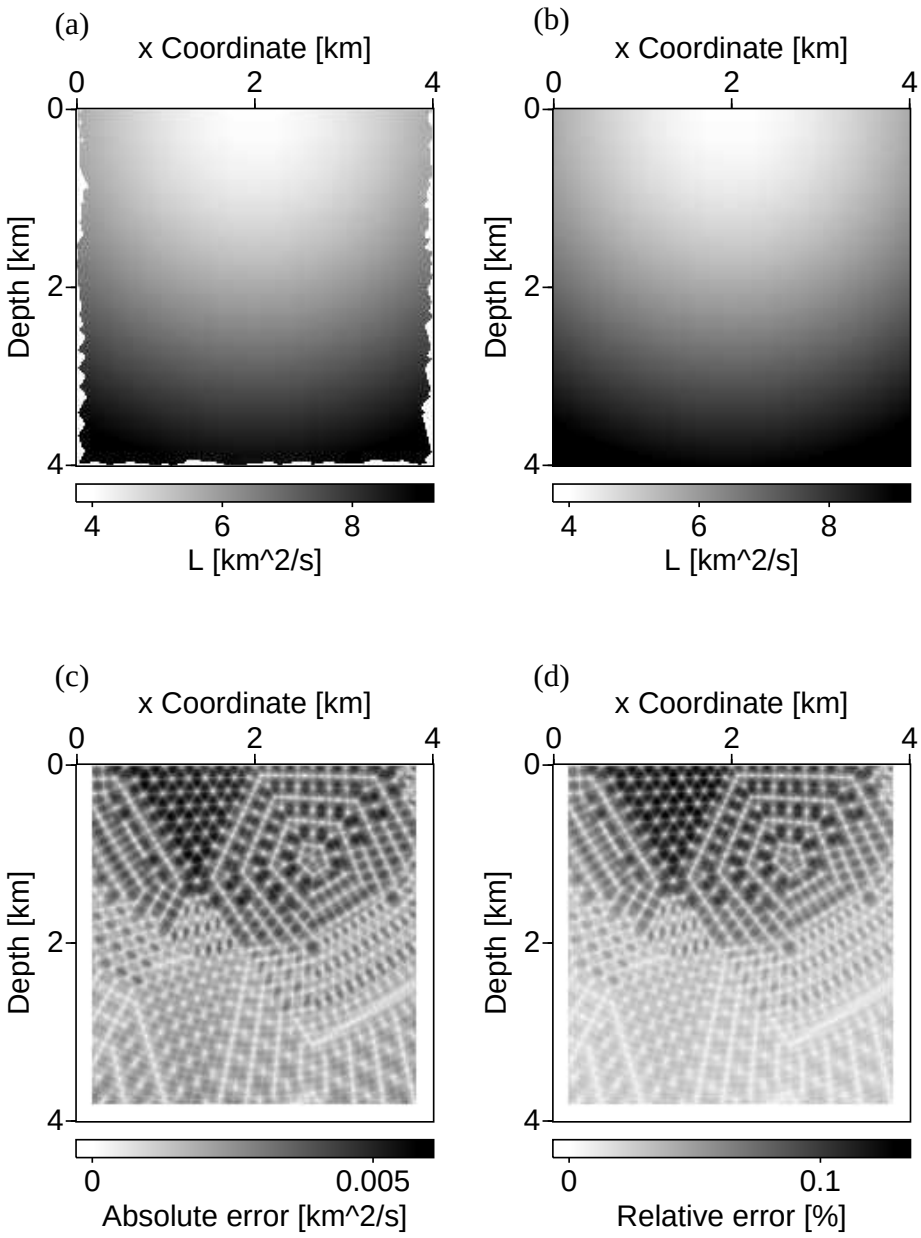

Figure 7.11: Relative geometrical spreading (L) in a homogeneous velocity model in the vertical section at $y = 2$ km. (a) computed by the WRT technique (b) analytical computation (c) absolute error (d) relative error. Note the pentagonal pattern due to the intersection of the rays with the vertical plane .

Constant Velocity Gradient Model

For the constant velocity gradient model, I show (1) the influence of the distance-dependent weights on the accuracy of computed traveltimes, (2) the influence of the insertion procedure for new rays on the accuracy of computed traveltimes, and (3) an accuracy test of the computation of the relative geometrical spreading.

To analyse the influence of the distance-dependent weights, I compute the traveltimes with and without distance weights (in the last case all weights are set to $1/12$). For the computation of traveltimes with the WRT technique, I use a time step of wavefronts of 70 ms, a UDT of 300 m and a time-difference threshold of 1 ms, corresponding to a change in wavefront curvature of maximum 4 m.

This example also shows the influence of the insertion procedure for new rays on the accuracy of computed traveltimes. This influence can be observed by analysing the distribution of the traveltime errors.

For the analytical computation of the traveltime τ at a point $P(x,y,z)$, I use the expression given by Dietrich and Cohen (1993),

$$\tau(x,y,z) = \frac{1}{b}\cosh^{-1}\left[\frac{x^2+y^2+\hat{z}^2+\hat{z}_0^2}{2\hat{z}\hat{z}_0}\right], \tag{7.11}$$

where $\hat{z} = v(z)/b$ and $\hat{z}_0 = v_0/b$. Note that the relation (7.11) is valid for each gridpoint in the model (no limitation with respect to turning points).

The wavefronts constructed from analytical traveltimes are shown in Figures 7.12a and 7.12b. The traveltime errors due to the constant-weighted averaging of extrapolated traveltimes (Figures 7.12c and 7.12d) are small along the rays and along the wavefronts, but the errors are large within the cells. The errors can be substantially reduced when using the distance-dependent weights (Figures 7.12e and 7.12f). In this example, the maximal traveltime error has been reduced by a factor of 20 (from 0.3 ms to 0.015 ms).

The distribution of the errors for traveltimes computed by the distance-weighted averaging of extrapolated traveltimes in the vertical slice defined by $x = 2$ km (Figure 7.12e) leads to three observations.

- The traveltime error increases with the increase of the difference in the wavefront curvature between adjacent rays. In this numerical example, this difference is large for small traveltimes.

- The traveltime errors are small along rays (less than 0.002 ms). The small errors prove that the chosen time step of rays leads to an accurate ray-tracing procedure.

- The insertion of a new ray reduces the traveltime errors. This is because the distances between the nodes and the gridpoints become smaller, and therefore the extrapolation of traveltimes is more accurate. However, it is necessary that the ray quantities of the inserted ray are accurate. In the WRT technique, a new ray is inserted by tracing it directly from the source. This insertion procedure maintains the accuracy of the new ray and therefore of the whole ray field in the model.

Figure 7.13a shows the relative geometrical spreading computed by the 3D WRT technique in a vertical section at $x = 2$ km. The exact computation of the relative geometrical spreading is shown in Figure 7.13b. The relative error in this vertical section is usually less than 1 %. Figure 7.14a shows the relative geometrical spreading computed by the 3D WRT technique in a vertical section at $y = 2$ km. The exact computation of the relative geometrical spreading is shown in Figure 7.14b. The relative error in this vertical section is usually less than 1 %.

Marmousi Model

To obtain the 3D version of the Marmousi model (Figure 7.15), I have extended the 2D unsmoothed Marmousi velocity model (Versteeg and Grau, 1991) in the third direction (x axis). By shifting the 2D velocity sections in the vertical direction, I simulate an anticlinal structure in the direction of the x axis. The ray tracing example is carried out after smoothing the 3D model such that spatial wavelengths lower than 200 m are removed. The velocity grid was resampled from 4 m to 20 m after smoothing. The source is located at the coordinates $x = 1.5$ km, $y = 6$ km, and $z = 0$ km.

The time step of rays was 10 ms, the time step of wavefronts 40 ms, the UDT 200 m, the LDT 50 m, and the time-difference threshold 4 ms. The maximum number of arrivals was three. Turning rays which carry later arrivals are removed. The computation with these parameters leads to 15816 rays. Wavefronts in three vertical sections normal to the x axis are shown in Figure 7.16.

Besides the traveltimes, I computed the take-off angles, the inclination angles in the model, and the relative geometrical spreading.

The upper picture in Figure 7.17 shows the inclination angle at source in a vertical section at $x = 1.5$ km. The distribution of the inclination angle shows several regions with smooth variations. At the border between these regions, the inclination angle has a discontinuity. The border between these regions coincides with a “kink” in the first-arrival isochrones, which indicates a folded wavefront. The lower two pictures in Figure 7.17 show the inclination and declination in a horizontal section at depth $z = 1.0$ km. Because the wavefront is not folded at this depth, the inclination angle shows a smooth distribution. The symmetry of the inclination angle with respect to $x = 1.5$ km results from the symmetry of the velocity model.

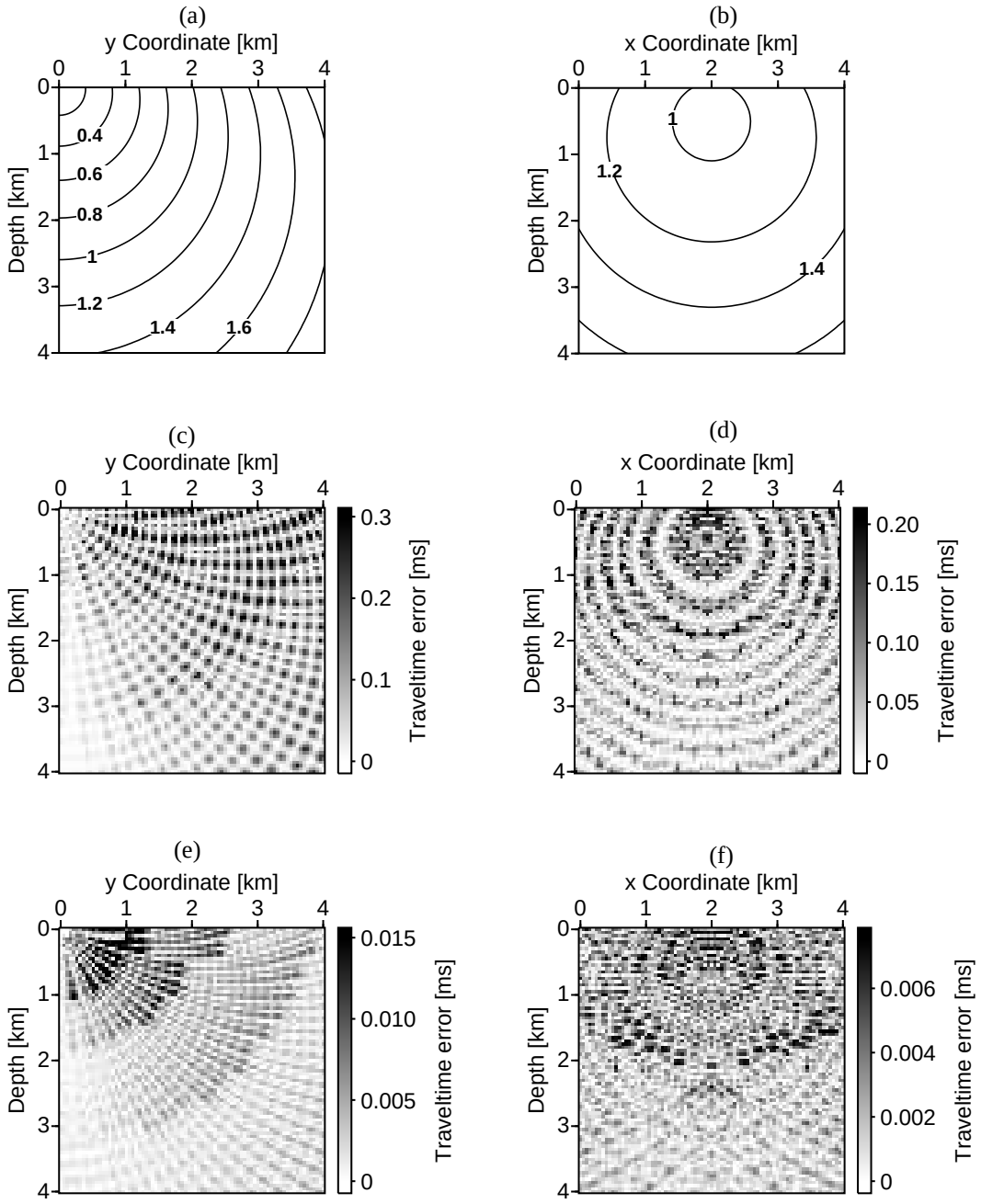

Figure 7.12: Results of the computation in the constant velocity gradient model in two vertical slices: (a), (c) and (e) for the vertical slice at $x = 2$ km; (b), (d) and (f) for the vertical slice at $y = 2$ km. (a) and (b): Wavefronts constructed from analytically computed traveltimes. (c) and (d): Traveltime errors due to the non-weighted averaging of extrapolated traveltimes. (e) and (f): Traveltime errors due to the distance-weighted averaging of extrapolated traveltimes. Please note the different error scales.

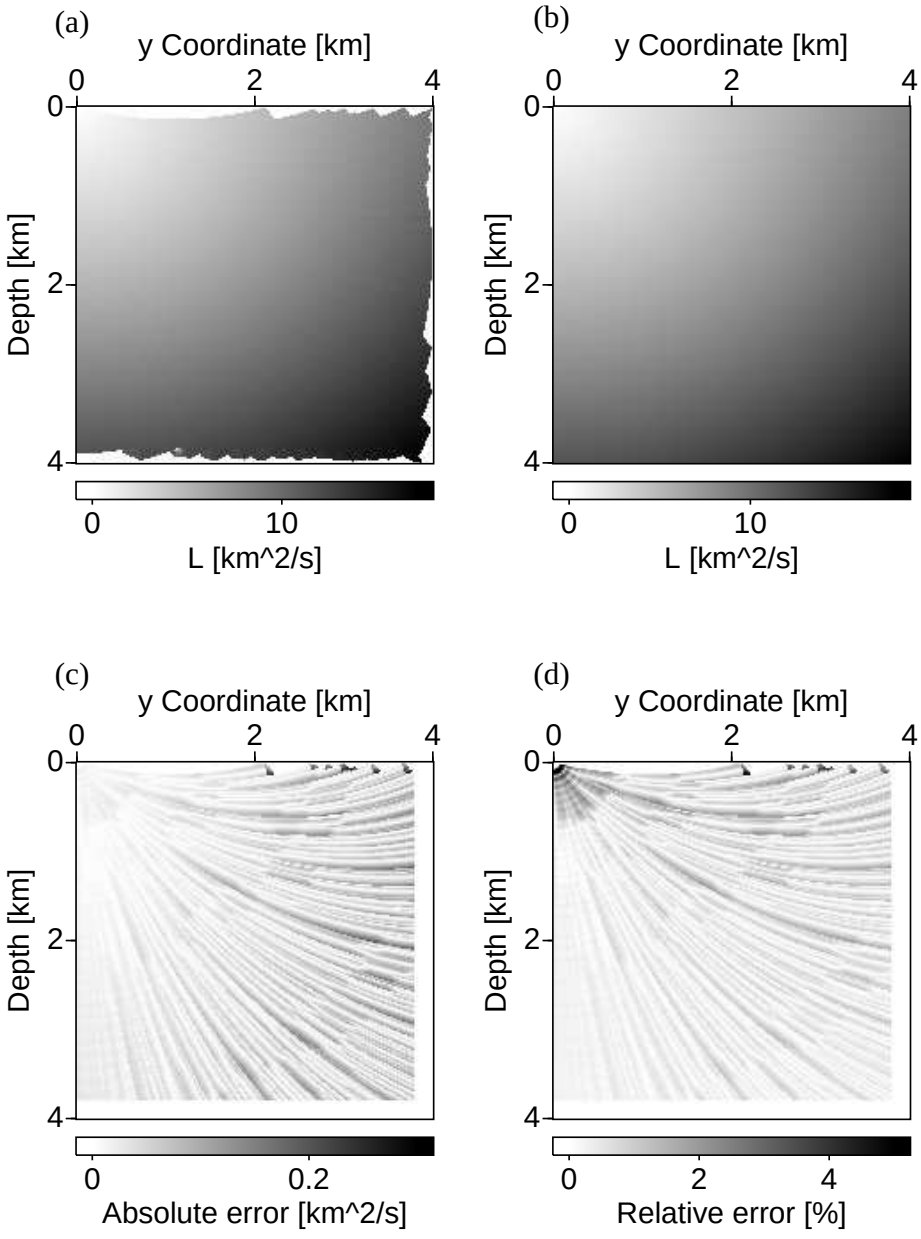

Figure 7.13: Relative geometrical spreading (L) in a velocity gradient model in the vertical section at x=2 km. (a) computed by the WRT technique (b) analytical computation (c) absolute error (d) relative error.

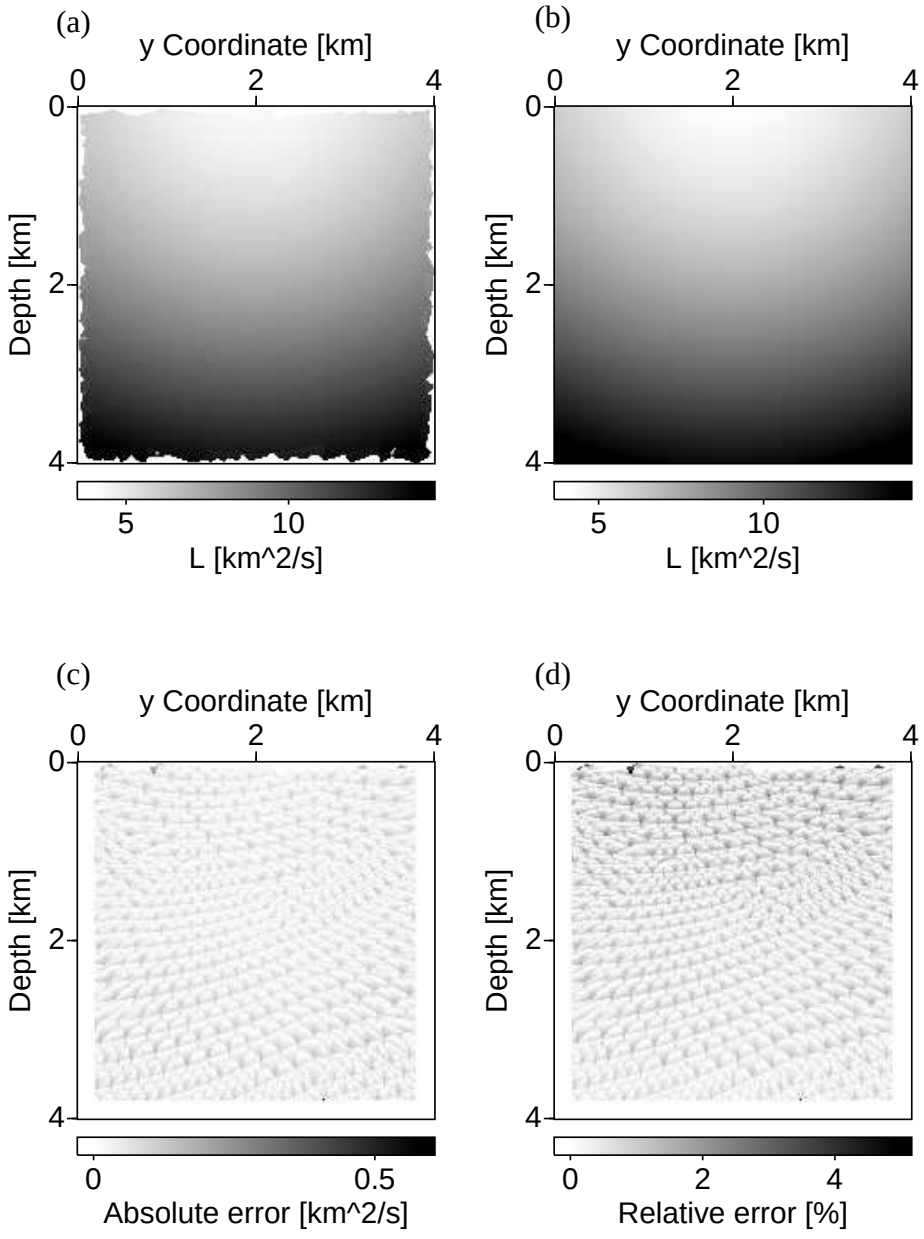

Figure 7.14: Relative geometrical spreading (L) in a velocity gradient model in the vertical section at y=2 km. (a) computed by the WRT technique (b) analytical computation (c) absolute error (d) relative error.

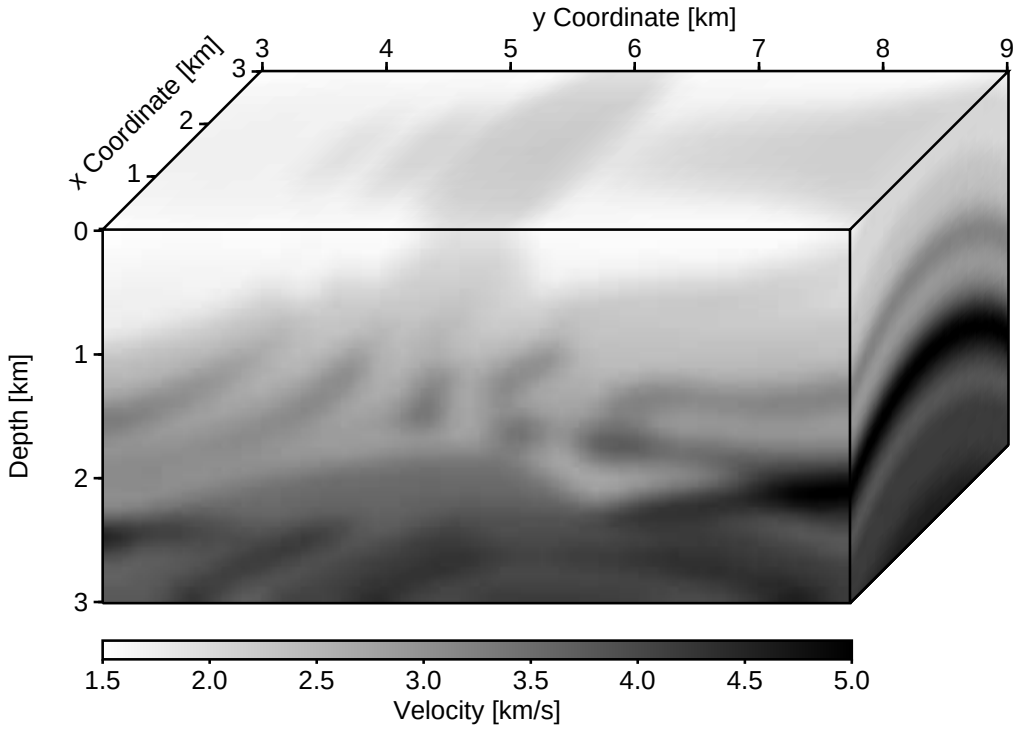

Figure 7.15: Smoothed version of the 3D Marmousi velocity model.

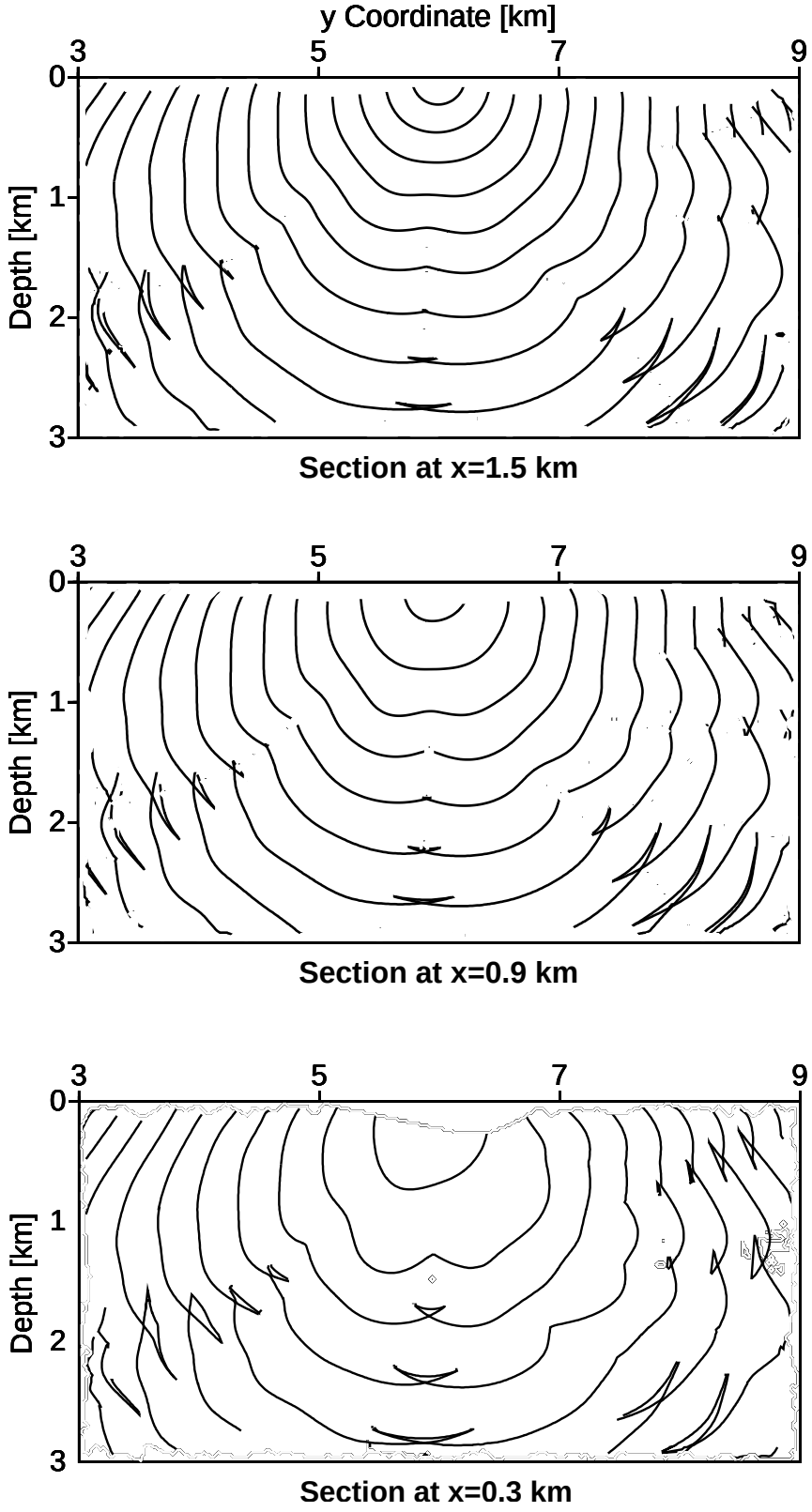

Figure 7.16: Wavefronts in three vertical sections. The distance between the wavefronts is 0.1 s.

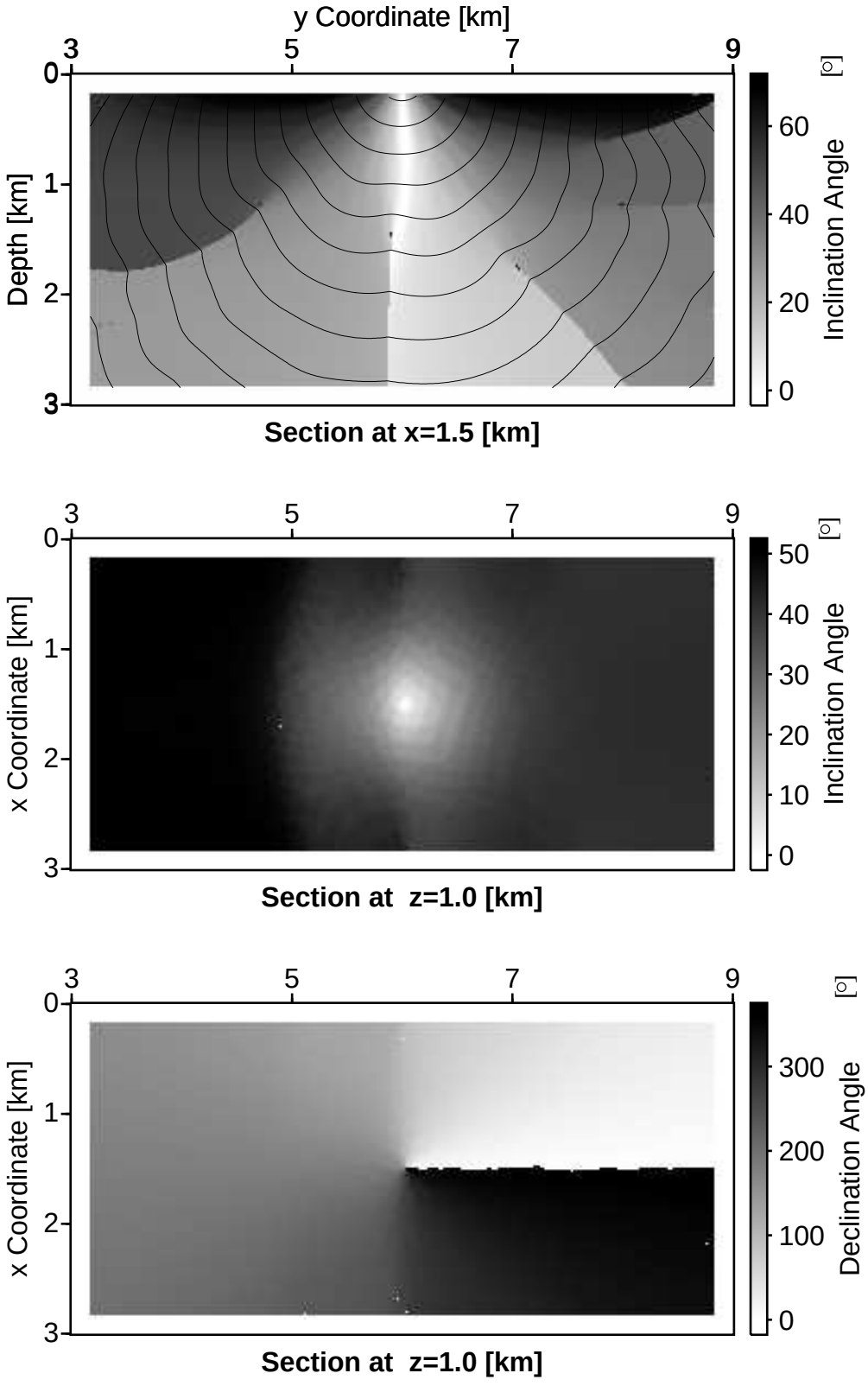

Figure 7.17: First-arrival inclination and declination angles at source in the 3D Marmousi model. The type of angle (inclination or declination) is indicated in the legend.

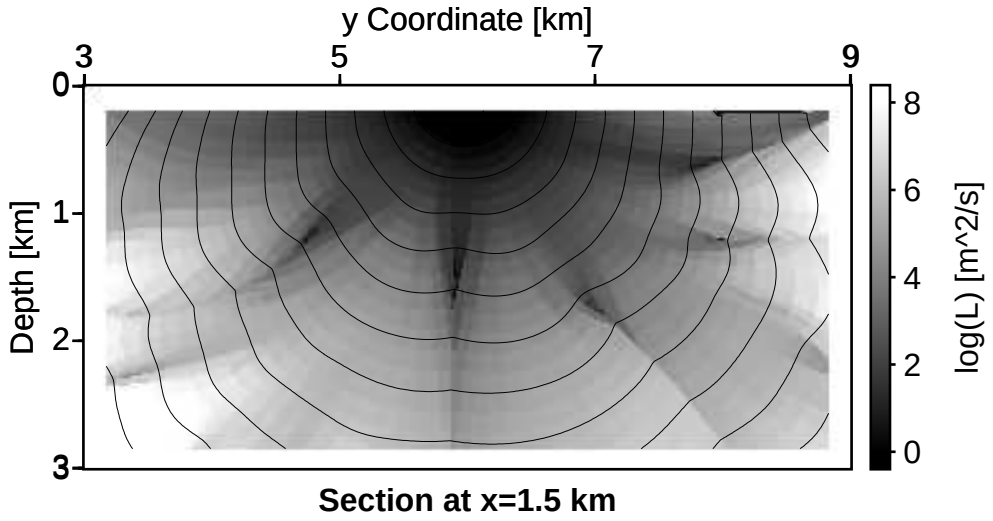

Figure 7.18: Relative geometrical spreading in a section normal to the x axis at 1.5 km.

Figure 7.18 shows the distribution of the relative geometrical spreading in the vertical section at $x = 1.5$ km. A low spreading value corresponds to large amplitudes.

The relative geometrical spreading can also be computed from traveltimes on a coarse grid (Vanelle, 2002; Vanelle and Gajewski, 1999). Figure 7.19 shows the differences between the relative geometrical spreading computed by the WRT technique (with a grid spacing of 20 m) and the spreading computed from traveltimes on a coarse grid. The grid distance of the coarse grid was 100 m. The spreading was subsequently interpolated with bilinear interpolation onto a 20 m grid. The two solutions coincide well apart from regions with discontinuous wavefronts where triplications occur.

There are two reasons for the above mentioned differences. Firstly, the algorithm which I use for the computation of spreading from traveltimes does not treat later arrivals, i.e., the interpolated spreading is not accurate in regions with discontinuous wavefronts (Vanelle, 2002). The consideration of later arrivals considerably increases the accuracy in these regions (Vanelle et al., 2003). Secondly, the WRT technique does not estimate traveltimes in caustic regions. Because at the beginning of a triplication the caustics are very close to the first arrivals, it may happen that first arrival traveltimes are not computed. Unfortunately, a single gridpoint without traveltime on the coarse grid will degrade the accuracy of the relative geometrical spreading for a large region (see, e.g., black square at $x = 7$ km and $z = 2$ km).

Figure 7.20 shows the distribution of the inclination angle in the model in the vertical section at $x = 1.5$ km. Kinks in the shape of first-arrival isochrones correspond to discontinuities in the distribution of the inclination angle. Inclination angles larger than 90°

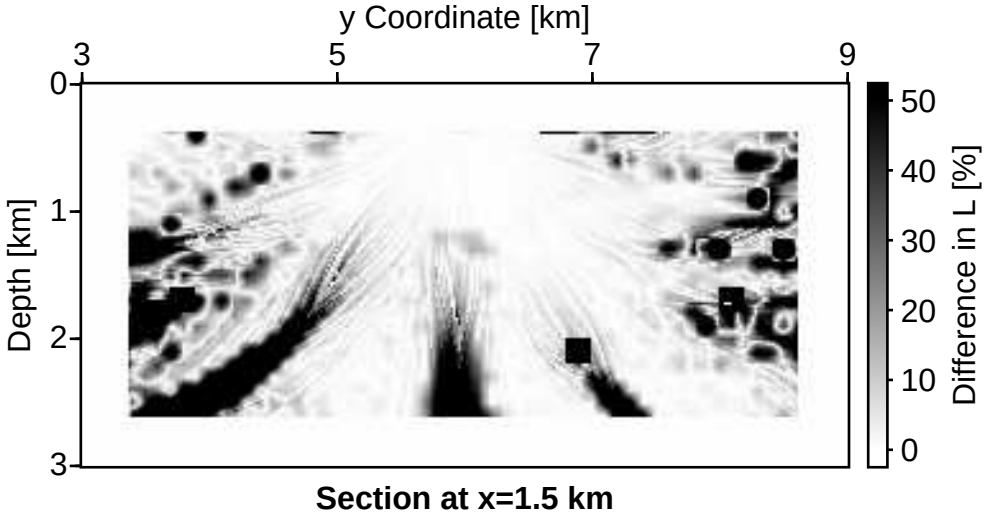

Figure 7.19: Comparison of spreading calculated by the WRT technique with spreading calculated from traveltimes on a coarse grid. The difference between both is given in percent. Regions where both results deviate are addressed in the text.

denotes diving rays. Except for the imaging of step salt structures, diving rays are usually not needed in PKDM. The WRT technique allows to dispense with the diving rays. The first-arrival isochrones in Figure 7.21 cover only the region without diving rays.

The decision to dispense with diving rays (detected by the sign of the vertical slowness component) considerably reduces the number of rays and increases the computational speed. If a diving ray is detected then (1) the computation of this ray is stopped, (2) the estimation of ray quantities within the cell bounded by this ray is stopped, and (3) no other ray in the neighbourhood of the diving ray is inserted. Table 7.1 shows the influence of later arrivals and diving rays on the number of rays.

In the next example, I show that the interpolation of new rays within the model leads to

No. of arrivals	LA diving rays	FA diving rays	No. of rays
3	Yes	Yes	24151
3	No	Yes	15816
1	No	Yes	9285
1	No	No	6549

Table 7.1: Influence of later arrivals and of the diving rays on the number of rays. LA=later arrivals; FA=first arrivals.

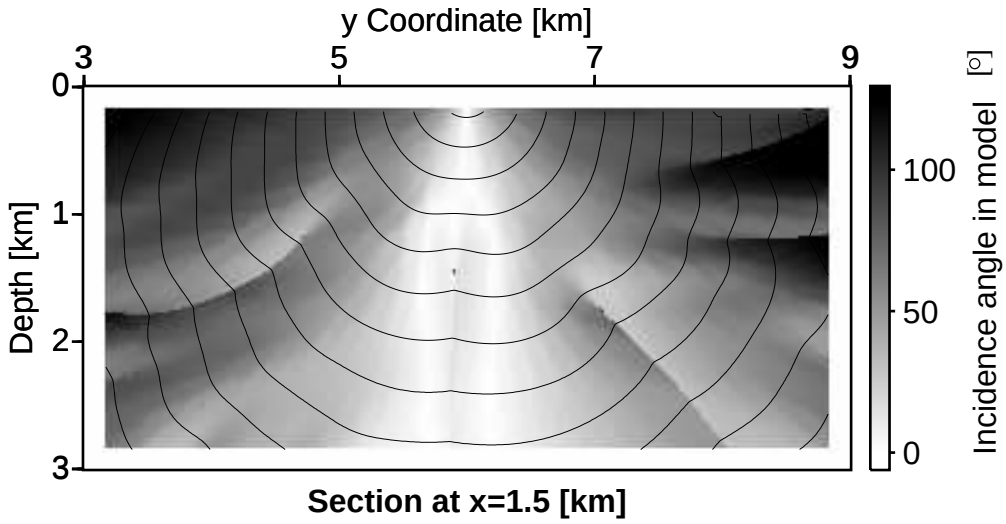

Figure 7.20: Inclination angle in the model in a section normal to the x axis at 1.5 km.

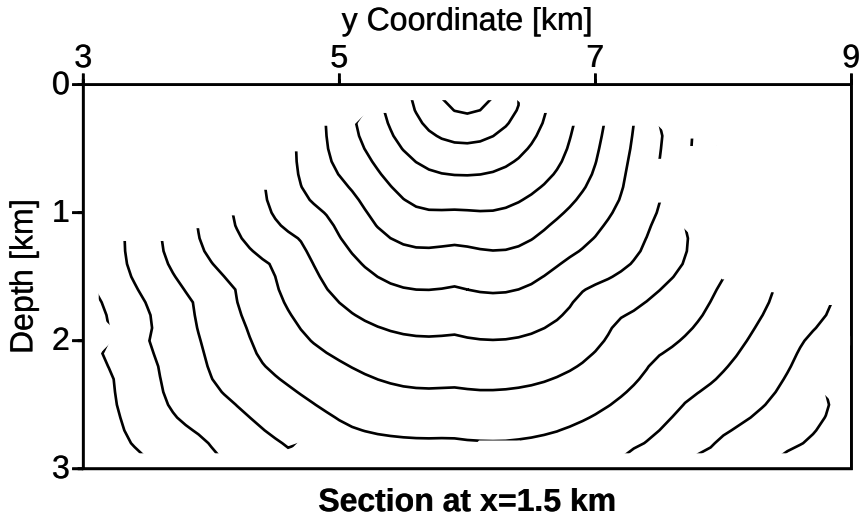

Figure 7.21: Isochrones of the first arrival traveltime without diving rays.

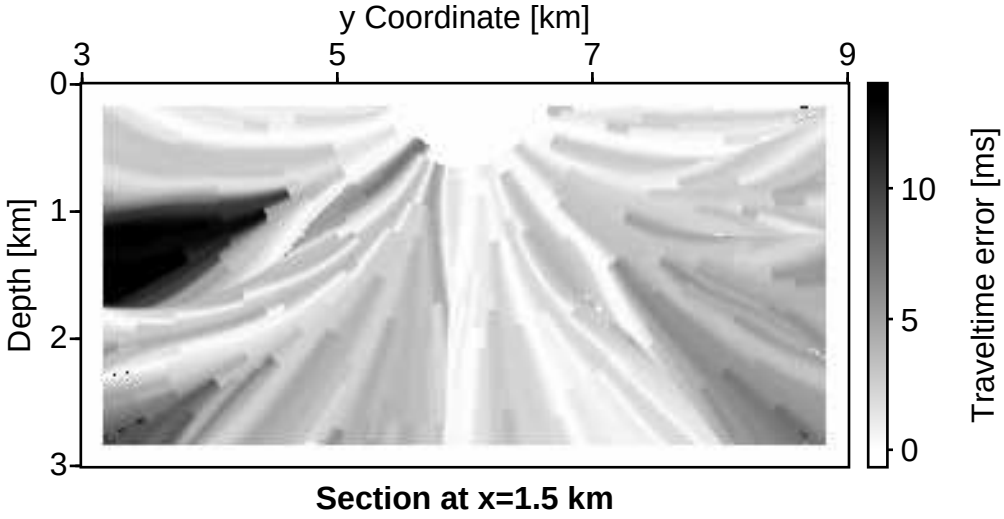

Figure 7.22: Traveltime errors due to the procedure for the insertion of new rays by interpolation within the model. The input parameters in the WRT technique and in the WRT/WFC hybrid method are the same. Reference traveltimes are computed by the WRT technique with 9285 rays (see text).

less accurate traveltimes than the insertion of new rays by tracing the rays directly from the source. For this example, I have replaced in the WRT technique the insertion procedure (which inserts new rays by tracing them from the source) by an insertion which inserts new rays by interpolation in the model (see Section 4.1.2 for the insertion procedure used in the WFC method). I call this mixture between the WRT technique and the WFC method the WRT/WFC hybrid method.

The traveltime differences between the traveltimes computed with the WRT technique and the WRT/WFC hybrid method are errors due to the WFC procedure for the interpolation of new rays.

The number of rays traced by the WRT technique was 9285, while the WRT/WFC method leads to 9001 rays. The CPU time for both computations was approximately the same (the WRT/WFC hybrid method was a little faster), but the traveltime errors due to the interpolation of new rays in the WRT/WFC method are up to 14 ms compared with the WRT technique (Figure 7.22). To reduce the maximum error, I have increased the ray density in the WRT/WFC hybrid method by using a maximum distance threshold of 100 m. The maximum traveltime error was reduced to 5 ms (Figure 7.23), but the number of rays increases to 37687. The larger number of rays and cells implies a longer CPU time for the WRT/WFC hybrid method.

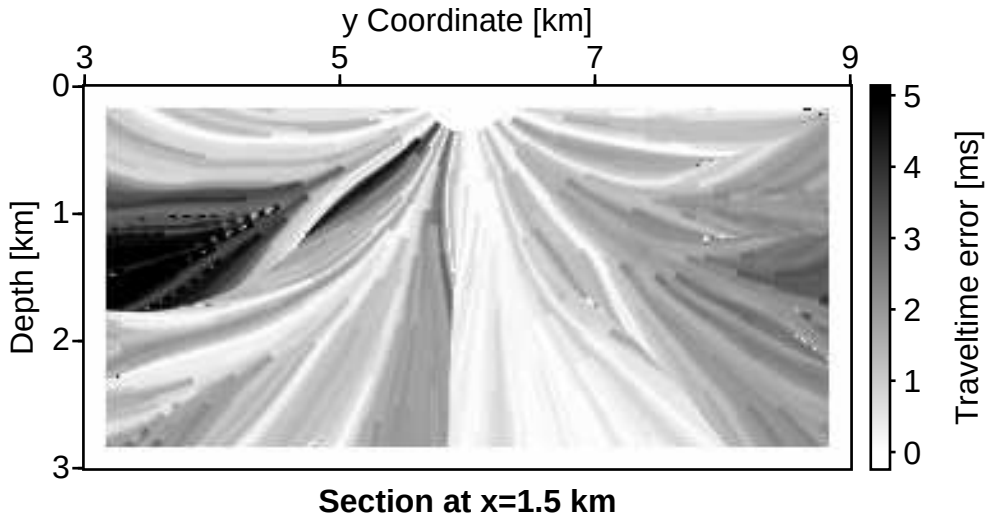

Figure 7.23: Traveltime errors due to the procedure for the insertion of new rays within the model. The input parameters in the WRT technique and in the WRT/WFC hybrid method are the same, but the maximum distance threshold differs (200 m in the original WRT, 100 m in the WRT/WFC method). Reference traveltimes are computed by the WRT technique with 9285 rays (see text).

7.4 Summary

The 3D WRT technique is based on the same concepts as the 2D WRT technique. Except for the estimation of traveltimes within cells, the extension from 2D to 3D is straightforward. Besides traveltimes, the 3D WRT technique allows the estimation of take-off angles, of inclination and declination angles in the model, and of the relative geometrical spreading. For the estimation of traveltimes within cells, I suggest a distance-weighted averaging of extrapolated traveltimes. Numerical examples in the homogeneous velocity model and in a constant velocity gradient model show a high accuracy of this approach. Numerical examples in the 3D Marmousi model show the applicability of the 3D WRT technique. Finally, I show that the insertion of a new ray by tracing it directly from the source not only increases the accuracy of the WRT technique but also increases the computational speed (the WRT/WFC hybrid method leads to 37687 rays, while the WRT technique leads to 9285 rays).

Chapter 8

Conclusions and Outlook

For the efficient computation of multivalued traveltimes, I have implemented a modified 3D WFC method and a 3D FDES-WFC method, and I have introduced the WRT technique (for abbreviations see Appendix B).

My implementation of the 3D WFC method is a combination of the methods proposed by Vinje et al. (1996a), by Lucio et al. (1996), and by Ettrich and Gajewski (1996). Additionally, I propose a new approach for the interpolation of a new ray on the wavefront. This approach has three steps: (1) interpolation of the slowness at the midpoint N between the two rays, (2) estimation of traveltimes by a second-order traveltime Taylor expansion at the point N, and (3) propagation of the new ray from the point N to the wavefront.

A WFC method can be used as stand-alone method for the computation of multivalued traveltimes for PKDM (present industry standard) or can be combined with a FDES to a hybrid method. In the FDES-WFC hybrid method, the first arrivals are computed by the FDES, while the later arrivals are detected, bounded and computed by the WFC method. Unfortunately, the advantage in terms of computational time is restricted to grids with small grid distances. Moreover, the hybrid method does not compute the arrivals between the wavefront crossing point and the focal point ("the second arrivals"). Because of this limitation, the hybrid method is not recommended in cases where the computation of all arrivals is important.

The WRT technique starts with few rays which are propagated stepwise through a smooth velocity model. To retain an optimal illumination by rays, I insert new rays by tracing them directly from the source. This insertion procedure avoids the main source of errors in the WFC methods (errors connected with the insertion of new rays by interpolation on the wavefront). In the WRT technique, I further reduce other types of errors. The insertion criteria reduce the errors connected with small-scale velocity variations and with caustic zones, while the distance-weighted averaging of extrapolated traveltimes reduces the errors of the traveltime estimation within large ray cells.

The insertion of new rays by tracing them from the source and the estimation of traveltimes by considering the wavefront curvature allows the use of large distances between rays, i.e., a gain in computational speed is achieved because fewer rays are used for the computation. In conclusion, the WRT technique is more efficient than the WFC methods.

The efficiency of the WRT technique strongly depends on the input parameters (e.g., the criteria for insertion of a new ray) which control the wavefront and ray densities. On the basis of traveltimes computed in a smoothed Marmousi model, I analysed these dependencies and suggested rules for an optimal choice of input parameters. With suitable input parameters, the WRT technique allows the computation of accurate traveltimes using a small number of rays and wavefronts.

The WRT technique has already found many applications. The traveltimes have been used for multivalued traveltime interpolation by the hyperbolic traveltime expansion (Dettmer, 2002; Vanelle et al., 2003), and for the location of seismic events (Robert Patzig, personal communication). The WRT technique was also extended to anisotropic media (Kaschwich and Gajewski, 2003). The WRT technique computes accurate traveltimes, which are needed for the traveltime based amplitude-preserving migration (Gajewski et al., 2002).

In future work, I want to integrate the 3D WRT technique in a prestack Kirchhoff depth migration environment which can handle multivalued traveltimes.

Appendix A

Notations

Deviations from the notations given below are mentioned in the text.

Matrix notation

Vectors are set in lowercase boldface letters, while matrices are set in boldface capital letters. To distinguish between 2×2 and 3×3 matrices, I shall use the symbol **(^)** above the letter for the 3×3 matrices. If the same letter is used for both 2×2 and 3×3 matrices, e.g., $\mathbf{M}$ and $\hat{\mathbf{M}}$, matrix $\mathbf{M}$ denotes the upper left submatrix of $\hat{\mathbf{M}}$,

$$\hat{\mathbf{M}} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix}, \qquad \mathbf{M} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}.$$

Similarly, I denote by $\hat{\mathbf{q}} = (q_1, q_2, q_3)^T$ the 3×1 column matrix and by $\mathbf{q} = (q_1, q_2)^T$ the 2×1 column matrix. The symbol T as superscript denotes the matrix transpose.

Component notation

The indices have the form of righthand suffices. The uppercase indices take the values 1 and 2, lowercase indices will take values 1,2,3. In this way, M_{IJ} denotes components of $\mathbf{M}$, M_{ij} components of $\hat{\mathbf{M}}$, $f(x_i) = f(x_1, x_2, x_3)$, $[f]_{x_k=0} = f(0,0,0)$ and $[f]_{x_K=0} = f(0,0,x_3)$.

Einstein summation convention

Summation convention applies over repeated suffixes, e.g.,

$$M_{ij}q_j = M_{i1}q_1 + M_{i2}q_2 + M_{i3}q_3.$$

Notation for partial derivatives

Partial derivatives with respect to Cartesian coordinates x_i and time t are signalled by a comma, e.g., $\lambda_{,i} = \partial\lambda/\partial x_i$, $u_{i,ji} = \partial^2 u_i/\partial x_j \partial x_i$, $u_{i,tt} = \partial^2 u_i/\partial^2 t$, $a_{mjkl,i} = \partial a_{mjkl}/\partial x_i$, etc.

Notation for scalar products

An arrow above letters denotes a 3D vector. The scalar product of two vectors $\vec{a}$ and $\vec{b}$ can be written either in vectorial notation as $\vec{a} \cdot \vec{b}$, in matrix notation as $\hat{\mathbf{a}}^T \hat{\mathbf{b}}$, in component notation as $a_i b_i$ or in the brackets notation as $\left(\hat{\mathbf{a}}, \hat{\mathbf{b}}\right)$.

Appendix B

Table of symbols

Symbols which occur locally may be explained in the text and do not appear in this list. Deviations from the below mentioned symbols are also mentioned in the text.

Abbreviations

2D	Two dimensional
3D	Three dimensional
DRT	Dynamic ray tracing
FD	Finite difference
HF	High frequency
KRT	Kinematic ray tracing
LDT	Lower distance threshold
PKDM	Prestack Kirchhoff depth migration
UDT	Upper distance threshold
WFC	Wavefront construction
WRT	Wavefront-oriented ray tracing

Indices

$\tilde{}$	In cases where the ray quantity is discontinuous at the reflection point, the tilde marks the quantity after reflection.
i, j, k, l, m	=1,2,3
s	S-wave
p	P-wave
(x)	Cartesian coordinate system
I, J	=1,2
G	Receiver (Geophone)
S	Source
T	Transpose matrix

Latin Lowercase

a_{ijkl}	Density normalised elastic tensor
c_{ijkl}	Elastic tensor
d	Distance
e	≈ 2.718282
f_i	Source term
$\hat{\mathbf{g}}^{(m)}(\gamma_i)$	Eigenvectors corresponding to the eigenvalue G_m
k	Number of caustics passed (KMAH index)
i	Imaginary unit, $\mathrm{i} = \sqrt{-1}$
$\hat{\mathbf{i}}^{(i)}$	Vector basis of the Cartesian coordinate system
$\hat{\mathbf{p}}(\gamma_i)$	Slowness vector
$\hat{\mathbf{n}}$	Normal vector to the wavefront
$\vec{u}(\vec{x}, t)$	Displacement vector

u	Displacement
u_i	Cartesian components of the displacement vector
s	Arclength
t	Time
$v(x_i)$	Phase velocity
w	Migration weight, Distance weight
$\hat{\mathbf{x}}(\gamma_i)$	Position vector
x_i	Cartesian coordinates

Latin Uppercase

F	HF analytical source signal
G	Receiver point (geophone)
G_m	Eigenvalue of the Christoffel matrix
$\mathcal{H}(x_i, p_i)$	Hamilton function
J	Jacobian of the transformation from γ_i to x_i
$\sqrt{\lvert J \rvert}$	Geometrical spreading
L	Relative geometrical spreading
$\mathcal{L}$	Normalised geometrical spreading
L_i, N_i, M_i	Vector operator
$\hat{\mathbf{Q}}^{(x)}$	Transformation matrix from γ_i to x_i
$\hat{\mathbf{P}}^{(x)}$	Transformation matrix from γ_i to p_i
U	Ray amplitude

Greek Symbols

δ_{ij}	Kronecker symbol
γ_i	Ray coordinates
γ_3	Parameter along the ray (e.g., τ or s)
γ_I	Ray parameters (e.g., take-off angles or inclination and declination)
θ_G, θ_S	Angle between the ray and the normal to the surface
ν	Parameter variable (e.g., τ or s)
$\lambda(x_i)$	Lamé elastic parameters
$\mu(x_i)$	Lamé elastic parameters
$\rho(x_i)$	Density
π	≈ 3.141593
τ	Traveltime or eikonal or phase function
ω	Angular frequency
Γ_{ij}	Components of Christoffel matrix
Ω	Ray

Appendix C

Geometrical spreading

The ray amplitude U for a wave under consideration (e.g., P wave) and in a medium without interfaces can be computed along the ray by solving an ordinary differential equation of the first order, the transport equation. The transport equation, which is obtained from equation (2.8), reads (Babich, 1961)

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\left(\sqrt{\rho J^{\tau}}U\right) = 0, \tag{C.1}$$

where J^{τ} is the Jacobian of the transformation from ray coordinates $\gamma_1, \gamma_2, \gamma_3 = \tau$ to Cartesian coordinates (Červený, 2001, p.202). For $\gamma_3 = s$ we have J^s and the relation between them is $J^{\tau} = vJ^s$, where v denotes the phase velocity. As a shortcut for J^s I use J. The quantity $\sqrt{|J|}$ is known as *geometrical spreading* (Červený, 2001, p.357).

The points where the function J vanishes are the *caustic points*. There are two types of caustic points. The caustic point of the first order (Figure C.1a) corresponds to a single root of the function J. At this point $J = \det \hat{\mathbf{Q}}^{(x)} = 0$ and $\operatorname{tr} \hat{\mathbf{Q}}^{(x)} \neq 1$. The caustic point of the second order (Figure C.1b) corresponds to a double root of J. At this point $J = \det \hat{\mathbf{Q}}^{(x)} = 0$ and $\operatorname{tr} \hat{\mathbf{Q}}^{(x)} = 1$, where $Q_{ij}^{(x)} = \partial x_i / \partial \gamma_j$. A caustic point of the second order only occurs in models of high symmetry and it is unlikely to be encountered in real earth models. However, in practical applications it may happen to have two consecutive first order caustics (Figure C.1c). At caustic points the zero-order ray theory is not valid because it predicts infinite amplitude.

The square root of J is taken as

$$\sqrt{J} = \sqrt{|\det \hat{\mathbf{Q}}^{(x)}|}\, \mathrm{e}^{-\mathrm{i}\frac{1}{2}\pi k}, \tag{C.2}$$

where k is known as the *KMAH index* or the *index of the ray trajectory* and represents the number of caustic points along the ray. Caustic points of the second order count two in this sum.

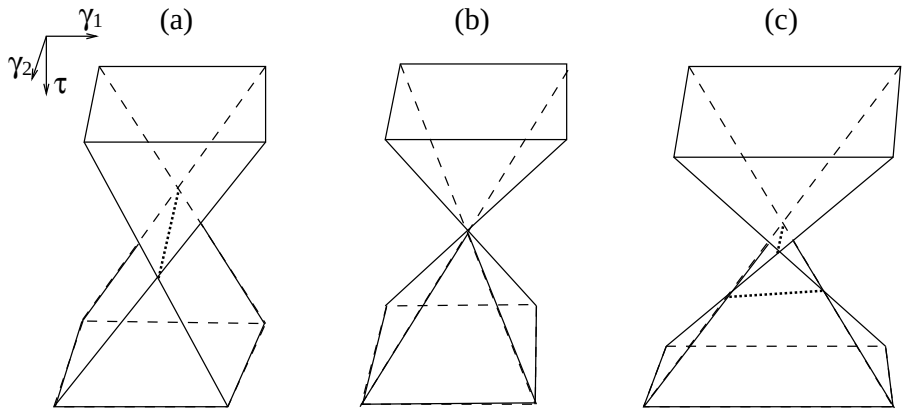

Figure C.1: (a) At a caustic point of the first order, the ray tube shrinks into an arc, perpendicular to the direction of propagation. (b) At a caustic point of the second order, the ray tube shrinks to a point. (c) Two consecutive first order caustics.

Sending three rays with the ray parameters (γ_1,γ_2), $(\gamma_1+\Delta\gamma_1,\gamma_2)$ and $(\gamma_1,\gamma_2+\Delta\gamma_2)$ and computing their positions $\hat{\mathbf{x}}^{(1)},\hat{\mathbf{x}}^{(2)},\hat{\mathbf{x}}^{(3)}$ on a given wavefront with KRT allows the approximate computation of J by the formula

$$J \approx \frac{\left|\left(\hat{\mathbf{x}}^{(2)}-\hat{\mathbf{x}}^{(1)}\right)\times\left(\hat{\mathbf{x}}^{(3)}-\hat{\mathbf{x}}^{(1)}\right)\right|}{\Delta\gamma_1\,\Delta\gamma_2}. \tag{C.3}$$

The *relative geometrical spreading* from source S to receiver G reads

$$L_{SG} = v_S\sqrt{\frac{|J_{SG}|}{\sin\theta_S}}, \tag{C.4}$$

where θ_S is the inclination angle at source and v_S is the velocity at the source. The relative geometrical spreading is reciprocal, i.e., $L_{SG}=L_{GS}$ (Červený, 2001, p.359).

Dividing the relative geometrical spreading by the velocity at the source point leads to the normalised geometrical spreading $\mathcal{L}$, which was introduced by Ursin (1990).

Appendix D

3D WRT technique: Numerical implementation

The WRT technique is very flexible. The accuracy and computational speed of this technique can be tuned by several input parameters. On the other hand, the numerical implementation is complex. To clarify the numerical implementation of the WRT technique, I show in Figure D.1 a flowchart with the main steps. In the following I describe these steps.

Read input file: The input file contains information about the model (file path, number of gridpoints, distance between gridpoints); the source (location; angular limitation of the initial direction of rays within a "radiation cone"); the propagation of the wavefront (time step of rays, time step of wavefronts); the estimation of ray quantities (distance between gridpoints, target region, number of multivalued arrivals, type of ray quantities); the end of the computation (maximum traveltime); the insertion of new rays (maximum-distance threshold, minimum-distance threshold, time-difference threshold); the path of output files.

Read velocity file: The velocity file is written as unformatted, binary data. The velocity should be smooth. The grid distance should be chosen in accordance with the degree of smoothing (Section 3.4).

Initialise rays/wavefronts: (Section 4.1.1). The first step is the initialisation of 12 rays which build the icosahedron. These 12 rays are then grouped into 20 ray tubes. Then the ray density is increased and the ray tubes are reorganised into an iterative process.

Propagate rays: The rays are propagated by a fourth-order Runge-Kutta method (Section 3.5). The time step of rays and the time step of wavefronts depends on the degree of smoothing (Section 6.3). The input quantities at arbitrary points are estimated by linear interpolation (Section 3.4).

Estimate ray quantities: A detailed description was given in Section 7.2.3.

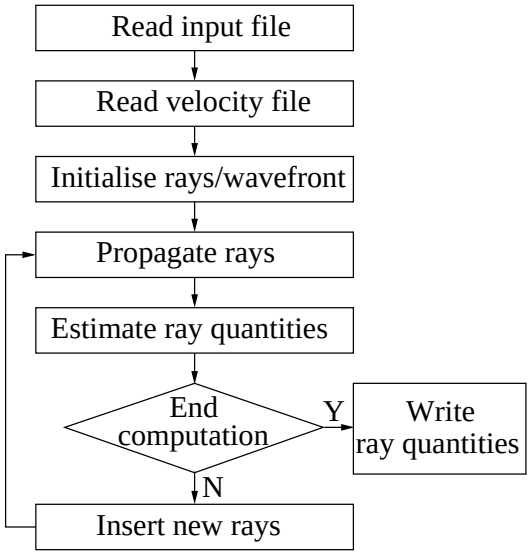

Figure D.1: Flowchart of the WRT technique.

End computation: The computation is ended if there is no active cell in the model, or if the maximum traveltime is reached, or if the maximum number of rays is reached.

Insert new rays: A detailed description was given in Section 7.2.2.

The complexity of the WRT technique is not only due to the high number of parameters which control the efficiency of the computation, but also due to the high number of variables which are needed for the computation. There are variables connected with rays, with ray tubes and with lateral tube sides. Some variables are used for storing ray quantities at the source, some other variables are computed and updated after each propagation of the wavefront.

In Table D.1, I show the quantities which are computed and/or stored along the rays, the ray tubes, and the lateral tube sides. The name of the entry (e.g., SlownessSource) indicates the ray quantity. The name is followed by a letter which indicates the type of quantity (**I** for integer, **S** for real single precision, **D** for real double precision). The number in parentheses indicates the number of columns in a two-dimensional array (e.g., the size of the array SlownessSource **D(3)** is [3, MaximumNumberOfRays]). The line indicates to which group the array is assigned (e.g., the SlownessSource is assigned to rays); the column indicates the location for which the quantity is stored.

The slowness at the source (SlownessSource) is needed for the estimation of the take-off angles (Section 7.2.4) and for the insertion of new rays (Section 7.2.2). Double precision is needed for the insertion approach.

Along the ray, I compute coordinates and slownesses. The computation of these quantities

	Source	Old wavefront	New wavefront
Rays	SlownessSource **D(3)**	SlownessOld **S(3)** CoordinatesOldNodes **S(3)** CoordinatesOldCorners **S(3)**	SlownessNew **D(3)** CoordinatesNew **D(3)** StatusRay **I**
Tubes	RaysInTube **I(3)** SolidAngle **S**	GeomSpreadingOld **S**	GeomSpreadingNew **S** NormalToWF **S(3)** StatusTube **I**
Sides			RaysInSide **I(2)** NewRay **I**

Table D.1: Quantities computed and stored along the rays, the ray tubes, and the tube sides.

with double precision requires a double precision storing on the new wavefront. Double precision is necessary for an accurate propagation of the ray through the velocity model. Beside the arrays mentioned in Table D.1, I use double precision for the estimation of the velocity and velocity derivatives at arbitrary points. The use of double precision for other variables is not necessary (e.g., the velocity and velocity derivatives at gridpoints are stored with single precision).

For the estimation of ray quantities within cells, single precision is sufficient. Before the rays are propagated, the coordinates and slownesses for the old wavefront are stored with single precision.

There are two types of coordinates for the ray at the old wavefront. One set of coordinates represents the coordinates of the ray at the nodes, the other set represents the corners of the cell. These coordinates are different for newly inserted rays. In order to ensure that the new cells are in contact with the old ones, the node of the newly inserted ray is projected onto the wavefront segment of the old cell. The approach is similar to the one in 2D and was described in Section 6.2.4.

The information about the status of the ray (e.g., inside or outside the model) is stored in the array StatusRay. Except for this array, all other above mentioned arrays are two dimensional. The indices for these two dimensional arrays are (3, MaximumNumberOfRays).

The connection between the number of a tube and the number of the rays which bounds this tube is stored in the array RayInTube(3, MaximumNumberOfTubes). The maximum number of tubes is twice the maximum number of rays. The solid angle at the source is needed for the computation of the relative geometrical spreading (Section 7.2.4). The information about the status of the tube (e.g., inside/outside the model, caustic cell) is stored in the array StatusTube.

A ray does not change its number during the whole computation. A ray tube does not

change its number as long as no ray is inserted within the tube. The insertion of new rays within a tube generates new tubes. The old tube is split up and one of the new tubes inherits the number of the old tube. The other new tubes receive new numbers. For more details, see Section 7.2.2. A tube side (usually) changes its number after each propagation step. This change is because the number of the ray sides is allocated dynamically after each propagation step. The dynamic allocation simplifies the numerical implementation and has practically no influence on the efficiency of the WRT technique.

The connection between the number of a ray side and the number of the rays which bound this side is stored in the array RaysInSide(2, MaximumNumberOfSides). The maximum number of sides is three times the maximum number of rays. The enumeration of the sides is important for the insertion of a new ray. A lateral side belongs to two adjacent tubes. The ray which is inserted within a lateral side is needed for the generation of new tubes in both adjacent tubes. The enumeration of the lateral sides is an efficient modality to "inform" the adjacent tubes about the new inserted ray. The number of the new ray is stored in the array NewRay.

Beside the arrays listed in Table D.1, the WRT technique uses four arrays with input quantities (one velocity and three velocity derivatives) and at least one array with output quantities (first arrival traveltime). The number of multivalued arrivals and the type of output quantities (e.g., relative geometrical spreading, slowness components) is defined by the user in the input file.

The memory requirement depends on the size of the model and on the degree of smoothing. For the Marmousi model (Figure 7.15) with a 200 m degree of smoothing, the memory requirement for the arrays listed in Table D.1 was about 5 MB (the maximal number of allowed rays was 25000). The memory requirement for the input quantities (velocity and velocity derivatives) is about 104 MB. Each of the four input files stores an array with the dimension $151 \times 301 \times 151$. The distance between gridpoints is 20 m. The memory requirement for an output quantity (e.g. the first-arrival traveltime array) was about 0.2 MB. Because the output grid is a coarse grid (the distance between gridpoints is 100 m), the grid dimension reduces to $31 \times 61 \times 31$. A maximum number of eight files may be stored for each arrival (the traveltime, the relative geometrical spreading, three components of the slowness in the model, three components of the take-off slowness). Storing all eight quantities for three arrivals leads to memory requirements of about 3 MB. In conclusion, the input quantities need 104 MB, the output quantities need 3 MB, auxiliary arrays need 5 MB. The large memory requirements for input quantities are connected with the use of linear interpolation. By using cardinal splines, the memory requirements for input quantities decrease from 104 MB to approximately 1 MB. The implementation of the WRT technique with cardinal splines and DRT may represent an alternative to the actual implementation.

Acknowledgments

Ich bedanke mich bei:

- Prof. Dr. Dirk Gajewski für die interessante Aufgabenstellung, für die Betreuung meiner Arbeit, für die vielen Ratschläge, die mir beim Verfassen dieser Arbeit sehr geholfen haben, und für die Unterstützung bei der Veröffentlichung von Teilaspekten dieser Arbeit.
- Dr. Ekkehart Tessmer für die Übernahme des Korreferats, für die stets kompetenten Antworten auf Fragen aller Art, und für die zuverlässig funktionierende Rechnernetzwerkumgebung.
- Dr. Claudia Vanelle für das sorgfältige Korrekturlesen meiner Arbeit und für die zahlreichen Tricks und Tipps aus der Linux-Welt.
- Dr. Henning Trappe und den Mitarbeiter der Firma TEEC für das Vertrauen und für die Unterstützung in der Endphase meiner Dissertation.
- Prof. Dr. Boris Kashtan für seine Diskussionsbereitschaft und die ausführlichen Erklärungen von diversen Aspekten der Strahlentheorie.
- Tina Kaschwich für das Korrekturlesen, für die gute Zusammenarbeit, und für die generell große Hilfsbereitschaft.
- Jan Dettmer für die sehr gute Zusammenarbeit und für das Erstellen der graphischen Benutzeroberfläche zu den WRT Programmen.
- allen Mitglieder der Applied Geophysics Group Hamburg, insbesondere bei Dr. Manfred Menyoli und Svetlana Soukina für anregende Diskussionen und für die freundschaftliche Atmosphäre.
- vielen anderen Personen die mich mit Rat und Tat während meiner Dissertation unterstützt haben. Ich erwähne hier unter anderen Dr. Christian Hübscher, Dr.

Jürgen Prüssmann, Dr. Tim Bergmann, Dr. Norman Ettrich, Dr. Stefan Buske, Dr. Ivan Pšenčik, Dr. Christian Hanitzsch, Dr. Einar Iversen, Dr. Gilles Lambaré, Thomas Hertweck. Weil die Liste nicht vollständig ist, entschuldige ich mich bei all den anderen Personen, die ich nicht namentlich erwähnt habe.

- Christel Mynarik und Birgit Stöfen-Vosberg für ermunternde Worte und für die Hilfe bürokratische Hindernisse erfolgreich zu bewältigen.

- den Organisationen, die mich während meiner Dissertation finanziell unterstützt haben. Dazu zählen die Europäische Union (EU), das Bundesministerium für Bildung und Forschung (BMBF), die Sponsoren des Wave Inversion Technology (WIT) Konsortium, und die Firma Trappe Erdöl Erdgas Consultant (TEEC) .

- meiner Frau Iuliana Coman und bei unserer Tochter Nora Coman für Liebe, Unterstützung, Verständniss und Motivation.

Bibliography

Aki, K., and Richards, P. G., 1980, Quantitative seismology, theorie and methods: W.H. Freeman & Co.

Åstebøl, K., 1994, Easy-to-use modelling – 3D ray field propagation in open ray models: EAEG/SEG summer workshop on construction of 3D macro-velocity models, Extended Abstracts, 114–117.

Babich, V., 1961, Ray method of calculating the intensity of wavefronts in the case of heterogeneous, anisotropic, elastic medium, *in* Petrashen, G., Ed., Problems of the dynamic theory of propagation of seismic waves: Leningrad University Press, 5, 36–46 (engl. translation: Geophys. J. Int., 118, 379–383, 1994).

Benamou, J., 1996, Big ray-tracing: Multivalue traveltime computation using viscosity solutions of the eikonal equation: J. Comput. Phys., **128**, 463–474.

Bevc, D., 1995, Imaging complex structures with first-arrival traveltimes: 65th Ann. Internat. Mtg, Soc. Expl. Geophys., Expanded Abstracts, 1189–1192.

Bulant, P., and Klimeš, L., 1999, Interpolation of ray theory traveltimes within ray cells: Geophys. J. Int., **139**, 273–282.

Bulant, P., Klimeš, L., and Žáček, K., 2001, Sobolev scalar products in the construction of velocity models: 63rd Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts, O–24.

Bulant, P., 1999, Two-point ray tracing and controlled initial-value ray tracing in 3-D heterogeneous block structures: J. Seismic Expl., **8**, 57–75.

Buske, S., and Kästner, U., 1999, A practical approach to the computation of multivalued geometrical spreading factors: 61st Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts, 4–16.

Červený, V., Molotkov, I. A., and Pšenčik, I., 1977, Ray method in seismology: Univerzita Karlova.

Červený, V., 1972, Seismic rays and intensities in inhomogeneous anisotropic media: Geophys. J. R. astr. Soc., **29**, 1–13.

Červený, V., 2001, Seismic Ray Theory: Cambridge University Press.

Clar, S., Ettrich, N., and Rühl, T., 1996, Can we image complex structures using smoothed macro-velocity models?: 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 543–546.

Coman, R., and Gajewski, D., 2000a, 3-D multivalued traveltime computation using a hybrid method: 70th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 2309–2312.

——— 2000b, 3D wavefront construction method with spherical interpolation: 62th Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts, C43.

Coman, R., and Gajewski, D., 2002a, 3-D wavefront-oriented ray tracing: Estimation of traveltimes within cells: 72st Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1923–1926.

——— 2002b, Wavefront-oriented ray tracing with optimal ray density: 64th Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts, C–020.

de Boor, C., 1978, A practical guide to splines: Springer-Verlag.

Dellinger, J., 1991, Anisotropic finite-difference traveltimes: 61st Ann. Internat. Mtg, Soc. Expl. Geophys., Expanded Abstracts, 1530–1533.

Dettmer, J., 2002, 3-D Laufzeitinterpolation in Kaustik-Regionen: Master's thesis, Universität Hamburg.

Dietrich, M., and Cohen, J. K., 1993, Migration to zero offset (dmo) for a constant velocity gradient: An analytical formulation: Geophysical Prospecting, **41**, no. 05, 621–644.

Eaton, D., 1993, Finite difference traveltime calculation for anisotropic media: Geophys. J. Int., **114**, 273–280.

Ettrich, N., and Gajewski, D., 1996, Wavefront construction in smooth media for prestack depth migration: Pure Appl. Geophys., **148**, 481–502.

Ettrich, N., and Gajewski, D., 1997, A fast hybrid wavefront construction - FD eikonal solver method: 59th Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts, E014.

Ettrich, N., and Gajewski, D., 1998, Traveltime computation by perturbation with FD-eikonal solvers in isotropic and weakly anisotropic media: Geophysics, **63**, 1066–1078.

Farra, V., and Le Bégat, S., 1995, Sensitivity of qP-wave traveltimes and polarisation vectors to heterogenity, anisotropy and interfaces: Geophys. J. Int., **121**, 371–384.

Gajewski, D., and Ψsenčik, I., 1987, Computation of high-frequency seismic wavefields in 3-D laterally inhomogeneous anisotropic media: Geophys. J. R. astr. Soc, **91**, 383–411.

Gajewski, D., Coman, R., and Vanelle, C., 2002, Amplitude preserving Kirchhoff migration: A traveltime based strategy: Stud. Geophys. Geod., **46**, 193–211.

Geoltrain, S., and Brac, J., 1993, Can we image complex structures with first-arrival traveltime?: Geophysics, **58**, 564–575.

Gray, S. H., Etgen, J., Dellinger, J., and Whitmore, D., 2001, Seismic migration problems and solutions: Geophysics, **66**, 1622–1640.

Hubral, P., and Krey, T., 1980, Interval velocities from seismic reflection time measurements: Soc. Expl. Geophys.

Kaschwich, T., and Gajewski, D., 2003, Wavefront-oriented ray tracing in 3d anisotropic media: 65th Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts, P041.

Kim, S., and Cook, R., 1999, 3-D traveltime computation using second-order eno scheme: Geophysics, **64**, 1867–1876.

Klimeš, L., and Kvasnička, M., 1994, 3-D network ray tracing: Geophys. J. Int., **116**, 726–738.

Kravtsov, Y., and Orlov, Y., 1990, Geometrical optics in inhomogeneous media: Springer Verlag.

Lailly, P., and Sinoquet, D., 1996, Smooth velocity models in reflection tomography for imaging complex geological structures: Geophys. J. Int., **124**, 349–362.

Lambaré, G., Virieux, J., Madariaga, R., and Jin, S., 1992, Iterative asymptotic inversion in the acoustic approximation: Geophysics, **57**, 1138–1154.

Lambaré, G., Lucio, P., and Hanyga, A., 1996, Two-dimensional multivalued traveltime and amplitude maps by uniform sampling of a ray field: Geophys. J. Int., **125**, 584–598.

Lecomte, I., 1993, Finite difference calculation of first traveltimes in anisotropic media: Geophys. J. Int., **113**, 318–342.

Leidenfrost, A., Ettrich, N., Gajewski, D., and Kosloff, D., 1999, Comparison of six different methods for calculating traveltimes: Geophysical Prospecting, **47**, 269–297.

Leidenfrost, A., 1998, Fast computation of traveltimes in two and three dimensions: Ph.D. thesis, University of Hamburg.

Liu, Z., 1994, A velocity smoothing technique based on damped least squares *in* Alkhalifah, T., Ed., in Project Review, CWP-153:: Colorado School of Mines.

Lucio, P., Lambaré, G., and Hanyga, A., 1996, 3D multivalued travel time and amplitude maps: Pure Appl. Geophys., **148**, 449–479.

Mispel, J., and Hanitzsch, C., 1996, The use of layered or smoothed velocity models for prestack Kirchhoff depth migration: 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 519–522.

Mo, L., and Harris, J., 2002, Finite-difference calculation of direct-arrival traveltimes using the eikonal equation: Geophysics, **67**, 1270–1274.

Moser, T., and Pajchel, J., 1997, Recursive seismic ray modelling: Applications in inversion and VSP: Geophysical Prospecting, **45**, 885–908.

Moser, T. J., 1991, Shortest path calculation of seismic rays: Geophysics, **56**, 59–67.

Nakanishi, I., and Yamaguchi, K., 1986, A numerical experiment on nonlinear reconstruction from first-arrival times for two-dimensional island arc structure: J. Phys. Earth, **34**, 195–201.

Operto, S., Xu, S., and Lambaré, G., 2000, Can we quantitatively image complex structures with rays?: Geophysics, **65**, 1223–1238.

Petrashen, G., and Kashtan, B., 1984, Theory of body-wave propagation in inhomogeneous anisotropic media: Geophys. J. R. astr. Soc, **76**, 29–39.

Podvin, P., and Lecomte, I., 1991, Finite difference computation of traveltimes in very contrasted velocity models: A massively parallel approach and its associated tools: Geophys. J. Int., **105**, 271–284.

Press, W., Teukolsky, S., Vettering, W., and Flannery, B., 1992, Numerical recipes in C: The art of scientific computing: Cambridge University Press.

Qin, F., Luo, Y., Olsen, K. B., Cai, W., and Schuster, G. T., 1992, Finite-difference solution of the eikonal equation along expanding wavefronts: Geophysics, **57**, 478–487.

Reshef, M., and Kosloff, D., 1986, Migration of common shot gathers: Geophysics, **51**, 324–331.

Reshef, M., 1991, Prestack depth imaging of three-dimensional shot gathers: Geophysics, **56**, 1158–1163.

Saito, H., 1989, Traveltimes and raypaths of first arrival seismic waves: Computation method based on huygens' principle: 59th Ann. Internat. Mtg, Soc. Expl. Geophys., Expanded Abstracts, 244–247.

Sava, P., and Fomel, S., 2001, 3-D traveltime computation using Huygens wavefront tracing: Geophysics, **66**, 883–889.

SchneiderW. A., J., 1995, Robust and efficient upwind finite-difference traveltime calculations in three dimensions: Geophysics, **60**, 1108–1117.

Sethian, J. A., and Popovici, M. A., 1999, 3-D traveltime computation using the fast marching method: Geophysics, **64**, 516–523.

Sheriff, R., and Geldart, L., 1995, Exploration seismology: Cambridge University Press.

Soukina, S., Gajewski, D., and Kashtan, B., 2001, Finite-difference perturbation method for 3D anisotropic media: 63rd Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts, P002.

Sun, Y., 1992, Computation of 2D multiple arrival travel time fields by an interpolative shooting method: 62nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1320–1323.

Symes, W. W., 1998, A slowness matching finite difference method for traveltimes beyond transmission caustics: 68th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1945–1948.

Thomson, C., and Gubbins, D., 1982, Three-dimensional lithospheric modelling at NORSAR: linearity of the method and amplitude variations from the anomalies: Geophys. J. R. astr. Soc., **71**, 1–36.

Ursin, B., 1990, Offset-dependent geometrical spreading in a layered medium: Geophysics, **55**, 492–496.

van Trier, J., and Symes, W. W., 1991, Upwind finite-difference calculation of traveltimes: Geophysics, **56**, 812–821.

Vanelle, C., and Gajewski, D., 1999, Determining geometrical spreading from traveltimes: 61th Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts, 1.56.

Vanelle, C., and Gajewski, D., 2002, Second-order interpolation of traveltimes: Geophysical Prospecting, **50**, 73–83.

Vanelle, C., Dettmer, J., and Gajewski, D., 2003, Detection of caustics and interpolation of later-arrival traveltimes: 65th Mtg., Eur. Assn. Geosci. Eng., Extended Abstracts.

Vanelle, C., 2002, Traveltime-based true-amplitude migration: Ph.D. thesis, University of Hamburg.

Versteeg, R., and Grau, G., 1991, Practical aspects of seismic data inversion: The Marmousi experience: 52th Mtg., Eur. Assoc. Expl Geophys., Proc. 1990 EAEG Workshop, 1–194.

Versteeg, R. J., 1993, Sensitivity of prestack depth migration to the velocity model: Geophysics, **58**, 873–882.

Vidale, J. E., 1988, Finite-difference calculation of travel times: Bull., Seis Soc. Am., **78**, 2062–2076.

Vidale, J. E., 1990, Finite-difference calculation of traveltimes in three dimensions: Geophysics, **55**, 521–526.

Vinje, V., Iversen, E., and Gjøystdal, H., 1993, Traveltime and amplitude estimation using wavefront construction: Geophysics, **58**, 1157–1166.

Vinje, V., Iversen, E., Åstebøl, K., and Gjøystdal, H., 1996a, Estimation of multivalued arrivals in 3D models using wavefront construction–Part I: Geophysical Prospecting, **44**, 819–842.

——— 1996b, Estimation of multivalued arrivals in 3D models using wavefront construction–Part II: Tracing and interpolation: Geophysical Prospecting, **44**, 843–858.

Vinje, V., 1997, A new interpolation criterion for controlling accuracy in wavefront construction: 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1723–1726.

Wang, L., 2000, Estimation of multi-valued Green's function by dynamic ray tracing and true amplitude Kirchhoff inversion in 3-D heterogeneous media: Ph.D. thesis, Colorado School of Mines.

Yilmaz, O., 2001, Seismic data analysis: Processing, inversion, and interpretation of seismic data: Soc. Expl. Geophys.